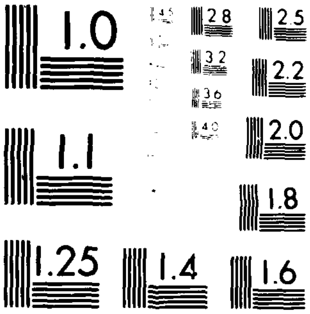
AD-A110 263   MINNESOTA UNIV  MINNEAPOLIS  DEPT OF ELECTRICAL ENGIN--ETC  F/G 9/5
ESSENTIAL PATTERN AND SEQUENCE SENSITIVITY IN SEMICONDUCTOR MEM--ETC(U)
JUL 80  A TUSZYNSKI  N00014-78-C-0741

UNCLASSIFIED  NL

1 OF 2
AD
A110263



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(12)

UNIVERSITY OF MINNESOTA

# ESSENTIAL PATTERN AND SEQUENCE SENSITIVITY IN SEMICONDUCTOR MEMORIES

A. TUSZYNSKI

ONR Contract N00014-78-C-0741

12

# ESSENTIAL PATTERN AND SEQUENCE SENSITIVITY IN SEMICONDUCTOR MEMORIES

A. Tuszynski

Electrical Engineering Department
University of Minnesota

ONR Contract N00014-78-C-0741

15 July 1980

## Contributors

The following researchers participated in the work which led to this report:

Daniel Burbank, Honeywell SSED

Jialin Jiang, Xian Jiaotong University

Mark Kroll, Intercomp

Jeffrey Kueng, U of M

Branislaw Vajdic, U of M

Gene Zemske, Control Data

Index

List of Figures

# Essential Pattern and Sequence Sensitivity in Semiconductor Memories

A. Tuszynski

## Abstract

Pragmatic classification of errors linked to a study of Essential Pattern and Sequence Sensitivity leads to a new and credible strategy for the validation of individual chips and entire memories. Three categories of errors are recognized: hard, non-random soft and random soft errors. Testing requirements for hard errors vary linearly with the number of storage bits. Exhaustive testing for NON-RANDOM soft errors is infeasible, because test requirements grow exponentially with the number of bits. And finally, individual testing of reasonably good chips for RANDOM SOFT ERRORS does not make any sense at all. That is why one must speak of validation rather than testing. Testing as such is addressed to hard errors only. Non-random soft errors are tracked-down by adverse analysis and diagnostics, to be eliminated in due course at the design and fabrication level. Random soft errors can be dealt with only by means of error correction circuitry. This may be expensive in terms of hardware.and access time but, once it is installed, the said circuitry will provide protection against all kinds of discrete errors.

Progress in the verification of VLSI devices is predicated on the availability of diagnostic instrumentation. Three promising techniques are being developed: 1) Stroboscopic Scanning Electron Microscopy 2) the Zero-Capacitance Probe and 3) Pseudo Analog Testing. The first has been on the horizon since at least

1968 and that is where it still appears to be. The second is well known as a relatively slow (50 kHz) electrometer; we have extended its response into the VHF range. The third, Pseudo Analog Testing, was developed by us from scratch; it is unwieldy but it often yields quantitative results where everything else fails.

More attention must also be given to the matter of process statistics. Worst case analysis cannot be executed without an appropriate data base. The performance margin of dynamic memories depends very directly on the uniformity of the fabrication technology. Many chips fail because of a single soft bit. Clarification of the nature of these failure would probably boost the yield and improve the realiability of memory chips.

Lastly, much remains to be done at the system level. We consider in appendix 2 error correction of the Hamming-code variety as a particularly efficient technique of fault-tolerent design and we discuss chip and system architectures for reduction of the risk of burst errors, but we feel, just the same, that block-oriented justification ought to be used in VLSI electronics.

All in all we advocate 1) conventional testing for hard errors attributable typically to stuck-at-faults, 2) extensive analysis and diagnostics for non-random soft errors produced more often than not by crosstalk manifested as Pattern or Sequence Sensitivity, and 3) block-oriented error correction for random errors induced most often by alpha particles and cosmic rays.

## 1.0 Nature of the VLSI-Test Problem

Rational approaches to VLSI testing are evolving disappointingly slowly. Many test engineers still cling to the "exhaustive testing" notion, failing to realize that exhaustive testing was nothing but an illusion even in the days of discrete components. In the VLSI era, it constitutes a veritable absurdity; the need for an "intelligent alternative to exhaustive testing" is pressing indeed [1].

What then is the nature of the quandary? Some will argue that there are more components but fewer access points then in the past; testing is, therefore, more difficult than it used to be. That is true enough, but accessibility can be regained through 1) design for testability in the Level Sensitivity scan fashion 2) the flip-chip packaging technique [3,4]. IBM has used both measures very successfully. One concludes, therefore, that accessibility alone does not present an insurmountable difficulty. Nor does the sheer number of components pose an unsurmountable barrier; modularity and hiererchical design take care of the numbers issue, provided only that the integrity of the interconnections as well as that of the modules is beyond doubt, before and after they have been placed on the chip.

There is more to the last proviso than meets the eye. VLSI is predicated on low power density and that translates into high impedance levels. VLSI also implies high-packing density and unavoidably high parasitic elements. The combination of high impedances and high parasitic elements brings about the real problem of VLSI.

The intrinsic problem of Large Scale Integration is crosstalk. With a million transistors on a die, it is impossible - at the present state of the art in circuit design and wafer fabrication - to guarantee freedom from parasitic interactions. To put it differently, we can't be sure that the funtionality of transistor r will not be upset by the data flow through transistor s. As a matter of fact, aboundant evidence of such crosstalk is readily available [5]. That is why one talks of Pattern Sensitivity, Sequence Sensitivity, Parasitic Oscillations, and what have you.

Parasitic oscillations are not discussed in this report; contrary to popular belief, they do occur occasionally in digital circuits - especially in ECL, as demonstrated in figure 1 - but classical forms of this problem are unquestionably analog in nature and are best discussed with reference to analog systems. Our immediate interest, on the other hand, is limited to undesirable peculiarities of digital circuits, Random Access Memories in particular. We begin with a classification of faults and errors, and follow through with an exposition of newly developed tools for the detections of the said peculiarities.

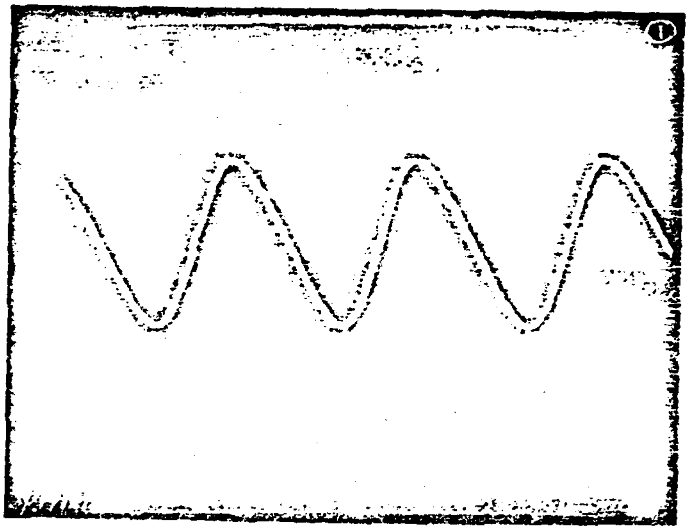


Fig.1: Parasitic Oscillations in Commercial ECL

## 2.0 Classification of Errors

Our classification of errors is intended to be pragmatic in nature. Presented in the fashion of figures 2a, 2b and 2c, it is supposed to simplify the exposition of techniques for the detection and correction of various errors. We postulate implicitely in figure 2a that a device is either good or bad - a suspect device is bad - but it may be good at one point in time yet bad at another; this is our definition of intermittence. To assert that a device is intermittent you need the outcomes of at least two indisputebly valid tests, the first showing the device to be bad and the second showing it to be good.

For the next step, we separate hard errors from soft, the latter being further subdivided into random and non-random errors. Hard errors are manifestations of permanent solid faults, classical examples being furnished by "stuck-at-faults" [6-9] attributable more often than not to shorts and breaks in interconnect matrices. Conversely, soft errors are obscure rather than solid. They can be random or non-random and, observed by the untrained eye, may be mistaken for symptoms of intermittent faults.

So much for our nomenclature. Proposed IEEE standards [10] speak of hard, soft and medium errors. "Hard errors are manifested by cells which will not properly store data under any condition of test operation". The term "soft errors" is reserved for "infrequent" random errors - alpha particle induced errors, for example - while the "medium" cognomen is supposed to cover all errors other than hard or soft.

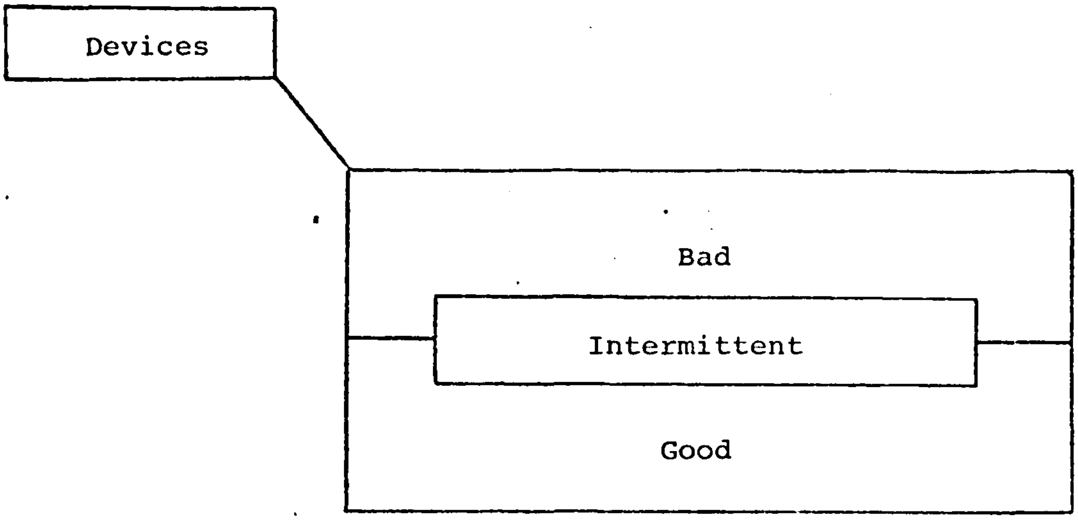


Fig. 2a: First-cut Classification of Devices

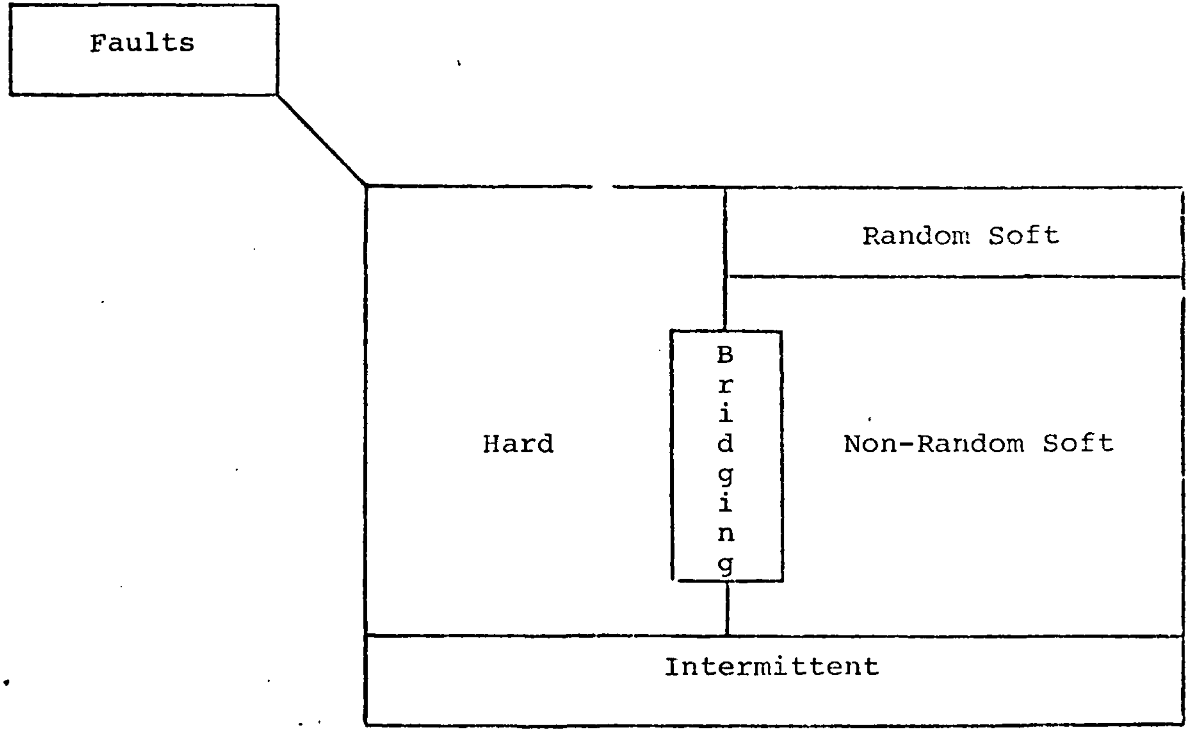


Fig. 2b: Classification of Faults

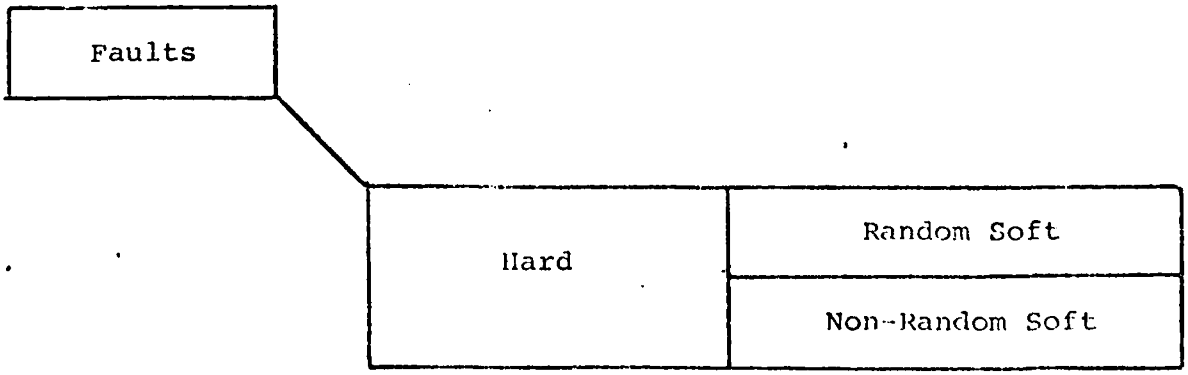


Fig. 2c: Simplified Classification of Faults

Whichever of the two languages is used, some difficulties will be encountered. Bridgind faults (see figure 2b) afford a conspicuous example, the word "bridging" implying unbidden physical coupling of two devices. They are solid enough, since nothing in a circuit could be more solid than a permanent short of two nodes - yet the symptoms may be confusing. That is why figure 2 shows them at the interface of hard and soft faults. Also troublesome are "open-gate" faults in MOS circuits and "open emitter" faults in bipolar TTL; both may display relatively slow parametric drift as well as fairly abrupt binary changes.

In spite of these complications, our classification of errors is simple and constructive. Fundamental is the separation of hard errors from soft and just as important is the distinction between random and non-random soft errors. In what follows, we speak mainly of hard errors, random soft errors and non-random soft errors (figure 2c).

## 2.1 Hard Errors

Hard errors are brought about by manufacturing flaws which are solid and discrete: a broken link in component "i" precipitates errors which are observable at the output of "i" and, more specifically, a ground-short at the base of transistor j induces errors observable at the collector of j. Crosstalk is specifically excluded from the notion of hard errors. That is why one arrives at a simple and convincing scheme of testing for hard errors. Checking a 1-by-N memory chip for hard errors one needs to demonstrate that:

1. there are indeed N distinct addresses, and
2. each cell can be switched from a) an unknown state to LOW, b) from LOW to HIGH, and c) from HIGH to LOW.

For this we need

$$x = 3\log_2 N \tag{1}$$

frames of N test each, for a total of

$$X = 3N\log_2 N \tag{2}$$

trials.

Increasing the memory capacity from N to 2N gets us, from equation 2

$$X(2N) = 3(2N)\log_2(2N) \tag{3a}$$

$$= 3(2N)[\log_2 N + \log_2 2] \tag{3b}$$

$$= 3(2N)[\log_2 N + 1] \tag{3c}$$

Comparing now 3c with 2, we find that

$$\frac{X(2N)}{X(N)} = 2[1 + \frac{1}{\log_2 N}] \qquad (4)$$

It follows without further ado that X(M) is roughtly proportional to M if M is reasonably large (say 128 or more). This - linked to the fact that hard errors are speed and temperature independent - means that testing for hard errors will never pose any undue difficulties, not in LSI, or VLSI or even ULSI. Trouble begins only when test requirements begin to translate into polynomial or, worse still, exponential functions of M; this we will see is the rule in the case of some classes of soft errors.

## 2.2.0 Soft Errors

The simplified version of our classification recognizes only hard errors and soft; intermittent and bridging errors are pushed into the background while errors camouflaged by intentional redundance are explicitely excluded from our considerations. Consequently, that which is not hard - i.e., solid and relatively easily observable - necessarily falls into the soft category. Soft errors are, therefore, vague and elusive, their manifestation depending on temperature, clock speed, environmental factors and data-flow patterns.

To resolve the vagueness of soft errors we begin with a basic question of uncertainty: are the observed phenomena reproducible? Can their existence be demonstrated on demand? Are they or are they not controllable? If the answers are affirmative, a deterministic regime prevails; if not, randomness is the order of the day. As shown in appendix 1 (reference 1) random errors can be separated from their non-random counterparts in spite of the obscure nature of soft faults.

## 2.2.1 Random Soft Errors

The common concept of randomness applies to the issue of errors. Errors are said to be random if they are sporadic and cannot be reproduced on a one-by-one basis; their time-average rate of incidence may or may not be controllable. Separation of random errors from deterministic mishaps (appendix 1) rests on the observation that an electronic chip does not remember incidental events indefinitely. The functionality of a chip in the soft-error sense (latch-ups are hard) may depend on events of the last microsecond, but it is, surely, independent from data processed a year ago. Our, as yet, unconfirmed estimate places the unintentional recall of Silicon Integrated Circuits at less than a minute. This being so, we can examine the character of every malfunction. Once an error is detected we interrupt the regular proceedings, returning to the state of 60 seconds ago, and then replay the events of the last minute.

The causes of random soft errors are usually found in the environment including the atmosphere on one hand [11], and the package in which the chip is housed on the other [12,13]. The effect of both can be reduced significantly by encapsulation of the chip [14-16], but complete elimination of radiation effects is absolutely impossible. One needs single-bit error correction [17] and one needs to do something about burst-errors. Chip and system architecture for the reduction of the risk of bursts is discussed in appendix 2 (reference 18). Testing of individual chips for random soft errors is pointless; one depends entirely on preventive measures and error correction.

### 2.2.2.0 Non-Random Soft Errors

Talking of non-random soft errors one refers willy-nilly to marginalities, that is performance marginalities attributable to circuit design flaws and fabrication-process fluctuation. After all, one talks of devices which work well under most circumstances; when they do fail, they fail in a deterministic fashion (by definition), though the actual mechanics of failure may be quite obscure. Obscure or not, the problem usually boils down to crosstalk manifested as sequence sensitivity or pattern sensitivity. The former refers to undesirable phenomena in the time domain while the latter puts the accent on topological peculiarities. Our present discussion is confined to Pattern Sensitivity. Sequence Sensitivity is examined in appendix 3 (reference 19).

### 2.2.2.1.0 Pattern Sensitivity

Examples of Ordinary Pattern Sensitivity are easy to come by. Consider, for example, the matter of access time. This parameter varies to a certain extent from chip to chip and, more importantly, from one address to the next. There exists a worst case condition, but identification of that condition may be difficult. What we are faced with then is a search for a key to the "worst case" situation. He who has the key can generate an efficient test procedure, but he who does not have it will probe in the dark talking of Pattern Sensitivity. Unduly optimistic specifications aggravate the problem, provoking arguments about marginal units, inflating the cost of testing and ultimately degrading system reliability.

Also to be considered is blatant abuse of the Pattern Sensitivity cognomen. Switching noise on external data buses and supply lines constitutes a case in point. The field engineer blames the vendor rather than himself for the cause of trouble, using Pattern Sensitivity as a convenient all-encompassing excuse.

The term Essential Pattern Sensitivity (EPAS) is, therefore, introduced to separate matters of processing and circuit design from superficial issues.

### 2.2.2.1.1.0 Essential Pattern Sensitivity (EPAS)

Essential Pattern Sensitivity can be viewed as a form of crosstalk which combined with performance marginalities results under certain circumstances in errors of binary character. It is said to exist when a chip which passes successfully one complete test can be shown to fail consistently another test. A chip which passes a comprehensive "checkerboard" test though it fails a "moving diagonal" test does display Essential Pattern Sensitivity (see appendix 3 for Schmoo plots of GALPAT and Column Disturb runs). Details of two examples of Essential Pattern Sensitivity are given below, the first to illustrate the fragility of the Dynamic RAM Sense Mechanism and the other to show that plain and simple design flaws do show-up in commercial devices.

The BIT-LINE circuitry of a 16k-RAM of the Mk4116 class is given in figure 3; its operation is described in appendix 2 and reference 20. Here we note that both sectors of the BIT-LINE are precharged to almost $V_{DD}$ (say $V_{DDP}$) during P2 at the beginning of a READ operation. The absolute value of $V_{DDP}$ is immaterial but the two sectors (left and right) of the BIT-LINE are supposed to be at the same potential. Some discrepancy will show up, however, because of differences between the precharge transistors $Q_{pl}$ and $Q_{pr}$. We refer therefore to $V_{Dl}$ and $V_{Dr}$ where for now

$$(V_{Dl} - V_{Dr})_p = \Delta V_p \qquad (5)$$

$\Delta V_p$ being referred to as the precharge offset. Another discrepancy arises because of leakage through the gating transistors (Qj,ect.); this descrepancy is pattern dependent. Roughly

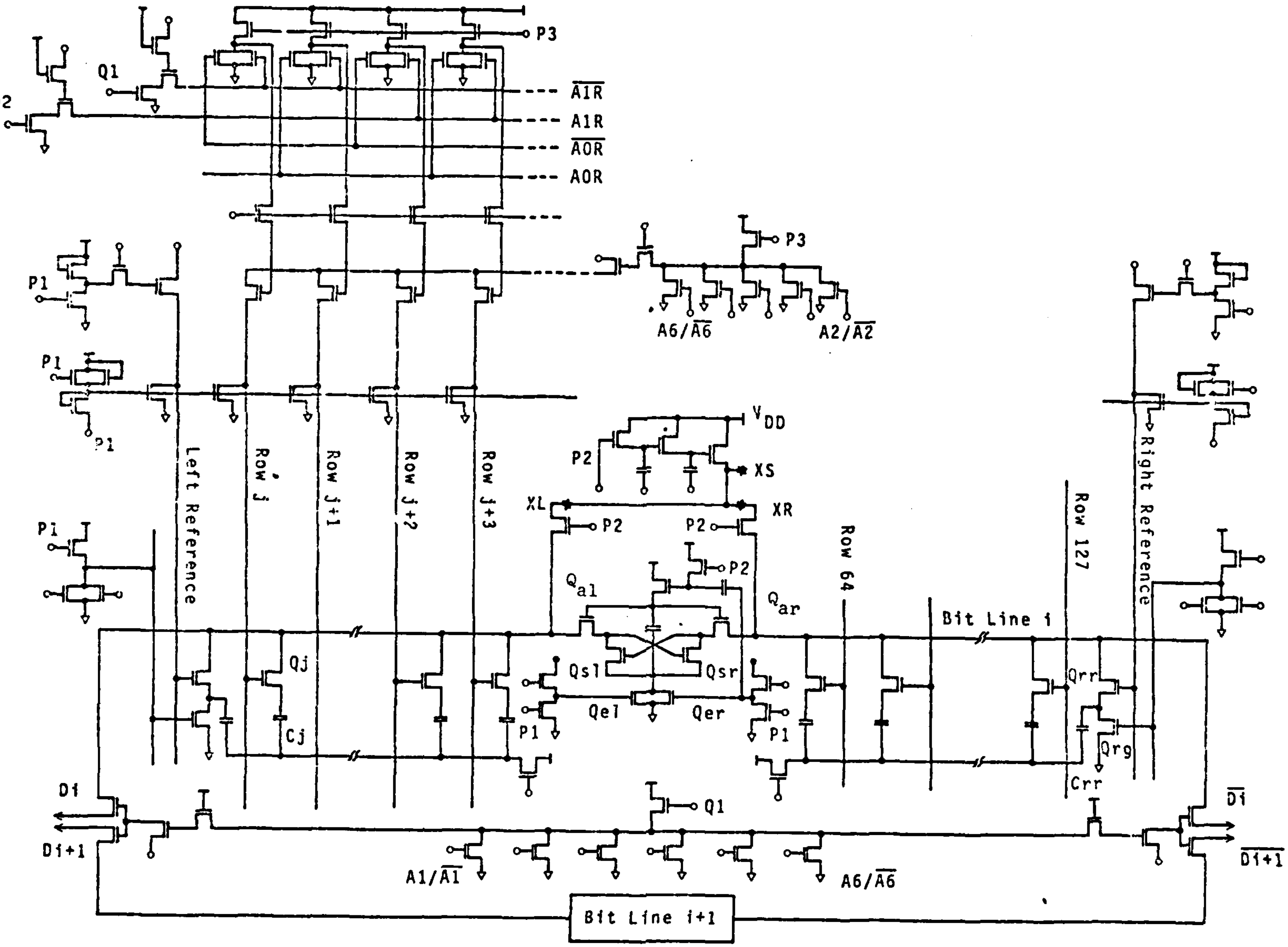


Fig.3: Fragment of BIT-LINE Circuitry of RAMs ā la MK4116

speaking, current leaks through a gating transistor only if the top plate of the corresponding storage capacitor is at a LOW potential. Thus with n LOWs on the left and m LOWs on the right, there will be a leakage offset $\Delta V_i$ of magnitude

$$\Delta V_i = (m - n)\Delta V_{ii} \tag{6}$$

where $\Delta V_{ii}$ stands for the average leakage-offset of single cell.

Yet another discrepancy, denoted by $\Delta V_o$, arises because of Sense-Amplifier offsets and Gating-Transistor inconsistencies. The total error ($\Delta V_e$) amounts, therefore, to

$$\Delta V_e = (V_{Dl} - V_{Dr})_e \tag{7a}$$

$$= \Delta\text{[illegible]} + \Delta V_o + (m - n)\Delta V_{ii} \tag{7b}$$

The nominal signal voltage produced in response to a left-side address is either

$$\Delta V_{sg}(H) = \frac{C_{rr}}{C_{br} + C_{rr}} V_{DD} \qquad \text{(figure 4)} \tag{8a}$$

$$\approx \frac{C_{rr}}{C_{bb}} V_{DD} \tag{8b}$$

or

$$\Delta V_{sg}(L) = - \frac{C_{sj}C_{br} - C_{rr}C_{bl}}{(C_{bl} + C_{sj})(C_{br} + C_{rr})} V_{DD} \tag{9a}$$

$$\approx - \frac{C_{sj} - C_{rr}}{C_{bb}} V_{DD} \tag{9b}$$

where $C_{bb} \approx C_{bl} + C_{sj} \approx C_{br} + C_{rr} \approx C_{bl} \approx C_{br}$

= Parasitic capacitance of the bitline (see figure 4)

and $C_{sj}$ = Capacitance of storage cell j

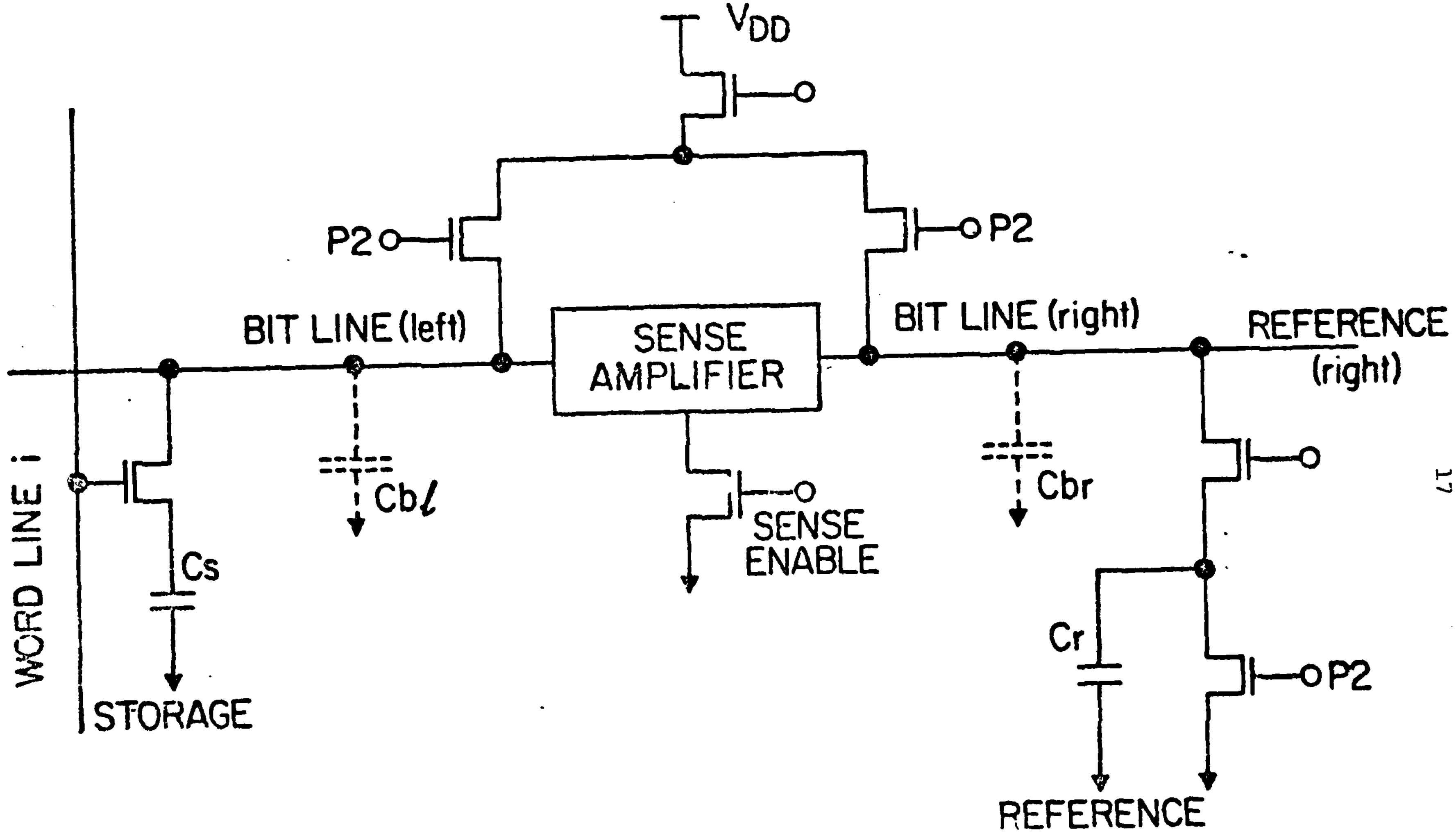


Fig. 4: Significant Capacitances

depending on whether a HIGH or a LOW voltage is being read: Comparison of equation 8b with 9b yields

$$C_{rr} = \frac{1}{2} C_{sj} \qquad (10)$$

as the optimum to storage ratio. It goes without saying that one wants the reference capacitors to be equal, i.e.,

$$C_{rr} = C_{rl} = C_r \qquad (11a)$$

and all storage capacitors to be equal too, that is

$$C_{sj} = C_{s(j + 1)} \ldots \qquad (11b)$$

$$= C_s \qquad (11c)$$

The ideal reference to storage ratio is, therefore, written as

$$C_{rr}/C_s = \frac{1}{2} \qquad (12)$$

with the warning that deviation from norm constitutes a particularly common and harmful complication. We write therefore, equations 8b and 9b as

$$|\Delta V_{sg}(L)| = |\Delta V_{sg}(H)| = \frac{C_s}{2C_{bb}} V_{DD} \qquad (13a)$$

introducing simultaneaously the capacitance-ratio-error

$$\Delta V_c = \frac{C_s - 2C_r}{C_{bb}} V_{DD} \qquad (13b)$$

Altogether, we have a signal - whose magnitude as given by equation 13a should be equal to .3 volts, if the numbers claimed by Intel [21] were to be believed - and four error components of which the first [$\Delta V_p(i)$] and third [$\Delta V_o(i)$] characterize an entire BIT LINE, the fourth {$\Delta V_c$(i left)} is associated either with one sector of a BIT-LINE (e.g., left sector of BIT-LINE "i") or an individual storage capacitor {$\Delta V_{cs}(j,i)$} and the third ($\Delta V_i$) is pattern sensitive. The worst case of $\Delta V_i$ occurs in patterns equivalent to the following example:

Location of the bit to be interrogated: Lower Sector, BIT-LINE "i"

Voltage at location to be interrogated: HIGH

Voltage at all other locations of the Lower Sector of
BIT-LINE "i": LOW

Voltage of all locations of the Upper Sector of BIT-LINE "i": HIGH

Leakage error: $$\Delta V_i(\text{w.c.}) = (\sqrt{N} - 1)\Delta V_{ii} \quad (14)$$

The pattern-sensitive leakage component may but need not be significant; everything in error analysis boils down to the signal-to-noise-ratio and the relative magnitudes of the individual components of noise. Both these issues are explored in more detail in section 4. There is abundant evidence to show that the above discussed Pattern Sensitivity is highly significant at elevated temperatures [22]. There is nothing one can do about it in short order. The user should, however, incorporate the worst case patterns into his hard-error high-temperature tests.

For an entirely different example of Pattern Sensitivity we turn to figure 5 and refernece 23. As is always the case with dynamic RAMS, one can READ, WRITE or REFRESH. The problems to be described are associated with the REFRESH operation. In this context consider two cycles numbered "n-1" and "n" respectively.

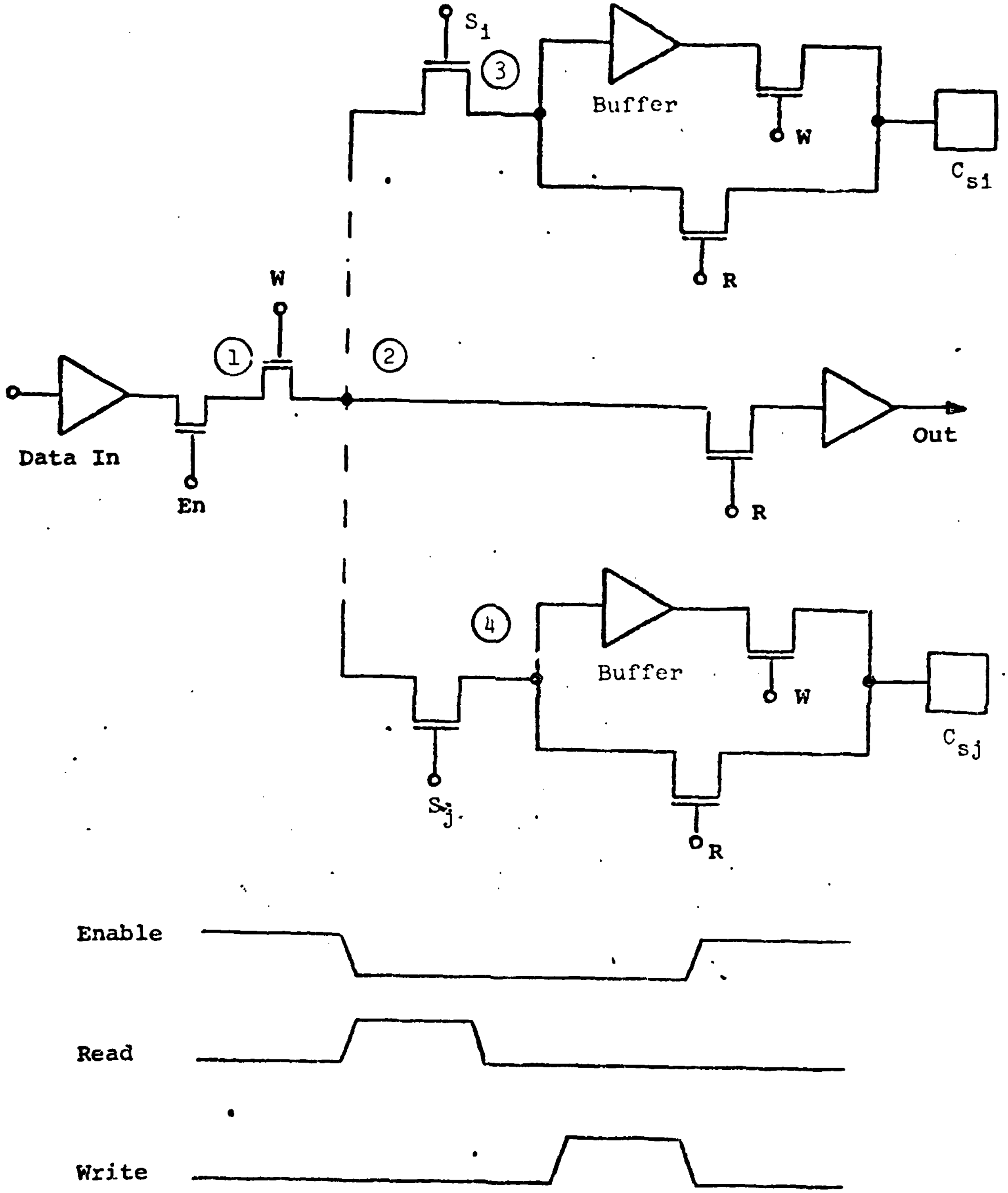


Fig. 5: BIT-LINE Detail of a Commercial 1k-RAM

First, READ storage cell j at cycle time "n-1" and next READ/REFRESH cell i at time "n". Finally, consider the sharing of charge by the capacitances of nodes 2 and 3 to observe that the outcome of the refresh operation on cell i may depend on the contents of cell j. This constitutes a case of Essential Pattern Sensitivity with a trace of Sequence Sensitivity.

Next, let data be written into any cell of time "n-1" and let any other cell be refreshed at time "n". Considering once more the parasitic capacitances of nodes 1 and 2, it becomes apparent that the outcome of the refresh operation of time "n" depends quite improperly on events which took place at time "n-1", and this gives us a transparent example of Essential Sequence Sensitivity.

We refer above to ESSENTIAL Pattern Sequence Sensitivities because we speak there of malfunctions which are directly related to circuit design problems, PASS GATE complication in particular. Availability of the PASS GATE in addition to the regular LOGIC GATE constitutes a valuable asset of the MOS transistor [24,25], well worth the price of the complications which, by the way, can be resolved quite easily.

One could argue, for that matter, that most Essential Pattern and Sequence Sensitivity mishaps are inexcussable. EPASS can be avoided or in the worst case, eliminated in redesign. One needs however adequate diagnostic tools as well as an adequate data base; diagnostic tools which facilitate node-by-node testing of VLSI chips and, a data base which offers statistical fluctuations as well as average values of process parameters.

Unchecked, Essential Pattern and Sequence Sensitivity begets catastrophics consequences. Since the functionality of the hardware may depend on the very data which is being processed, test requirements become utterly unrealistic.

### 2.2.2.1.1.1 The Incidence of Errors as a Function of the Data Which is Being Processed

A fault will only be detected by an appropriate set of test vectors. If, for example, input A in the AND GATE of figure 6 is physically shorted to ground (stuck at zero) nothing abnormal will be detected at the output terminal by means of test vectors A = 0, B = 0; A = 1, B = 0; or A = 0, B = 1. The existence of a stuck-at-zero problem can be disclosed only by the A = 1, B = 1 vector. Speaking of combinatorial logic and testing in general, one can assert that a fault which changes the transfer function of a circuit from

$$f_c = f_1(x_1, x_2, \ldots, x_n) \tag{14a}$$

to

$$f_c = f_2^{\alpha}(x_1, x_2, \ldots, x_n) \tag{14b}$$

will be divulged by test vector

$$a = \{a_1, a_2, \ldots, a_n\} \tag{15a}$$

$$= \{x_1, x_2, \ldots, x_n\} \tag{15b}$$

only if

$$f_1(a) + f_2(a) = 1 \tag{16}$$

Thus, manifestation of faults is always data dependent in the sense that appropriate test vectors must be applied if a particular fault is to be disclosed. There is however more to data-dependence than the matter of test vectors. As we have shown, the very functionality of a ciruit may depend on the data which is being processed by it: a device which passes an exhaustive

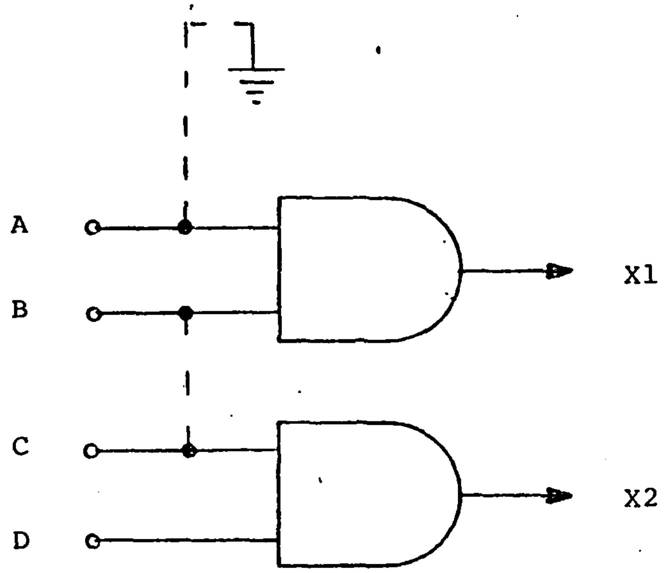


Fig. 6: Combinatorial Logic, Potential "Stuck-at" and "Bridging" Faults

(as far as hard errors are concerned) GALPAT test may well be rejected by a COLUMN DISTURB test. What is exhaustive in the hard-error domain is by no means exhaustive in the soft-error context. Therein lies a crucial distinction between hard and soft errors. Essential Pattern and Sequence Sensitivity is the principal cause of this distinction.

From now on we will simply say that hard errors are data independent while their soft counterparts do depend on the flow of data in terms of both Pattern and Sequence Sensitivity. Furthermore the word "data", as used here, refers to addresses, operands and unclassified alphanumerics.

### 2.2.2.1.1.2 Numerical Assesment of Pattern Sensitivity

Overlooking issues of sequence sensitivity we state that the contents of a binary N-bit memory can be arranged into

$$\Xi = 2^N \tag{17a}$$

$$= 10^{.3N} \tag{17b}$$

patterns.

To check all of these, in the case of a 1-by-N chip, one must go through at least

$$\Omega = N\Xi \tag{18a}$$

$$= N\times 10^{.3N} \tag{18b}$$

test. Operating at the rate of a million trials per second, i.e., 3.6 × $10^9$ trials per hour or about 3 × $10^{13}$ test per year, we would need

$$X(64k) = \Omega/3x10^{13} \tag{19a}$$

$$= 2x10^{19,000} \tag{19b}$$

years to test a 64k-RAM. In one hour we can test exhaustively for Pattern Sensitivity a memory of no more than a total of 40 bits. This is the practical limit in exhaustive Pattern Sensitivity Testing even if, by exclusion of Sequence Sensitivity, we limit ourselves to the simplest form of the pattern problem.

This does not mean that one does not test for Pattern Sensitivity. All we need to give up is the concept of exhaustive testing. We analyze, diagnose and eventually test in accordance with our worst-case-scenerio.

## 3.0 Numerical Summary of Test Requirements

As outlined in section  we need at least

$$\Omega(h.e.) = 3N\log_2 N \tag{21}$$

to check for hard errors. Allowing for crosstalk of the nearest-neighbor type, one boosts the test requirements to

$$\Omega(n.n.) = 24N\log_2 N \tag{22}$$

In both of the above tests there is essentially a linear relationship between the test requirements and the capacity of the memory chip. However, a three-halves relationship results, i.e.,

$$\Omega(c.\sqrt{N}) = k_1 N^{3/2} \tag{23}$$

if crosstalk between all bits in a row or a column or on a diagonal is consider. Simple crosstalk between any two bits leads to the N-SQUARE law,

$$\Omega(c.N) = k_2 N^2 \tag{24}$$

and finally, Pattern Sensitivity brings forth the exponential law,

$$\Omega(p.s) = Nx2^N \tag{25}$$

Needless to say, practical testing of VLSI devices must be limited to the linear cases. That means hard errors, proximity effects and identified worst case scenarios of the Pattern and Sequence Sensitivity category.

## 4.0 Diagnostic Tools

High packing density is predicated on low power density and that, in turn, leads to high impedances and susceptibility to external loads. VLSI circuits, MOS devices in particular, cannot be tested by means of conventional test techniques. New tools are required and three different types are becoming available. The first, the Strotoscopic Scanning Electron Microscope (26,27) is being developed by Bell Laboratories, IBM, Siemens and Fujitsu. The projected price-tag is $200k, but the long-awaited instrument is still not available. The second, a microprobe characterized by a very low input-capacitance has been available for some time in various audio-frequency versions (28,29); we have extended the response of these probes into the VHF/UHF range. The cost of our probe falls below the $500 mark. The third, a technique named Pseudo Analog Testing has been developed by us from scratch.

## 4.1 The Zero Capacitance Probe

Dealing with large circuits, one can't overemphasize the importance of power economy; the concept of VLSI is predicated on low power-density: with a million transistors on the chip just one micro-ampere per transistor results in 5 watts at 5 volts, and that calls for water cooling, an inacceptable demand for general-purpose work, current practice in low-density ECL and 3D not withstanding. Consequently there is a trend towards low-voltage concepts (IIL, ISL, etc.) in the Bipolar World and sub-micro-ampere techniques in MOS. Our MICROPROBE is designed for MOS work.

Where currents are low, impedances are high. The BIT-LINE of a 16k dynamic RAM is equal to a stand-alone 1 pF capacitor over a major part of an operating cycle. Conventional oscilloscopes are useless. We need a probe whose input impedance is no larger than .1 pF and we need a response of 100 MHz to boot. Difficult as these requirements may be, our probe foots the bill.

The original schematic diagram of our probe is shown in figure 7a. What we have there is complete follower: the emitter follows the gate in the usual fashion and the collector follows, in turn, the emitter, via Q4 and the Zener diode. Thus one gets neutralization of the collector-gate capacitance as well as the emitter-gate capacitance. A discrete transistor breadboard of figure 7a displays an input capacitance of .8 pF and a band-width of almost 100 MHz at a total current drain of 20 mA.

Figure 7b shows the latest version of our MICROPROBE. Two items merit attention: 1) the GATE-TO-SUBSTRATE capacitance is highlighted and neutralized while 2) the external base resistances

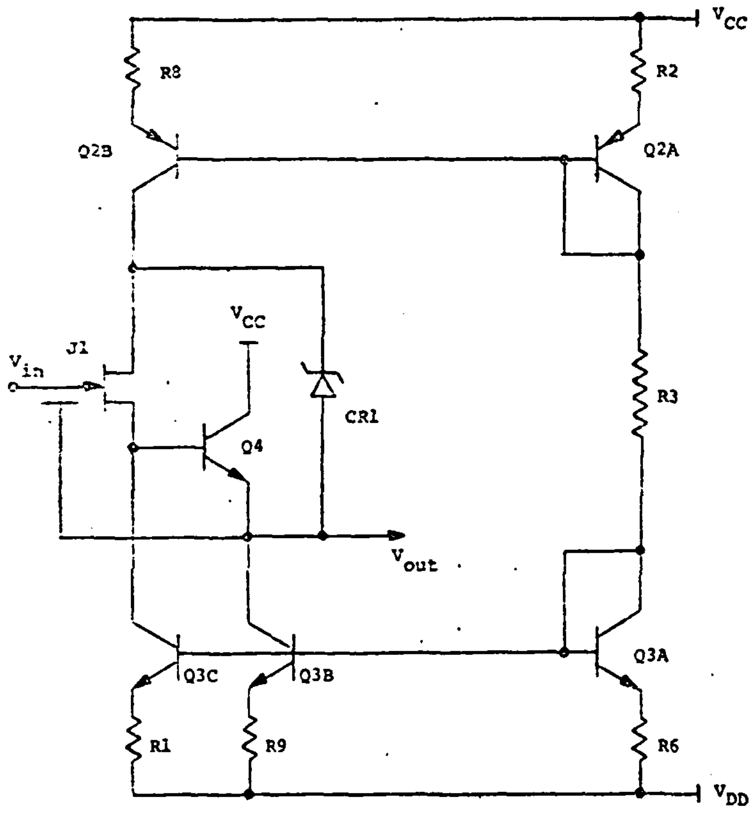


Fig. 7a: Schematic Diagram of Microprobe JJ-I

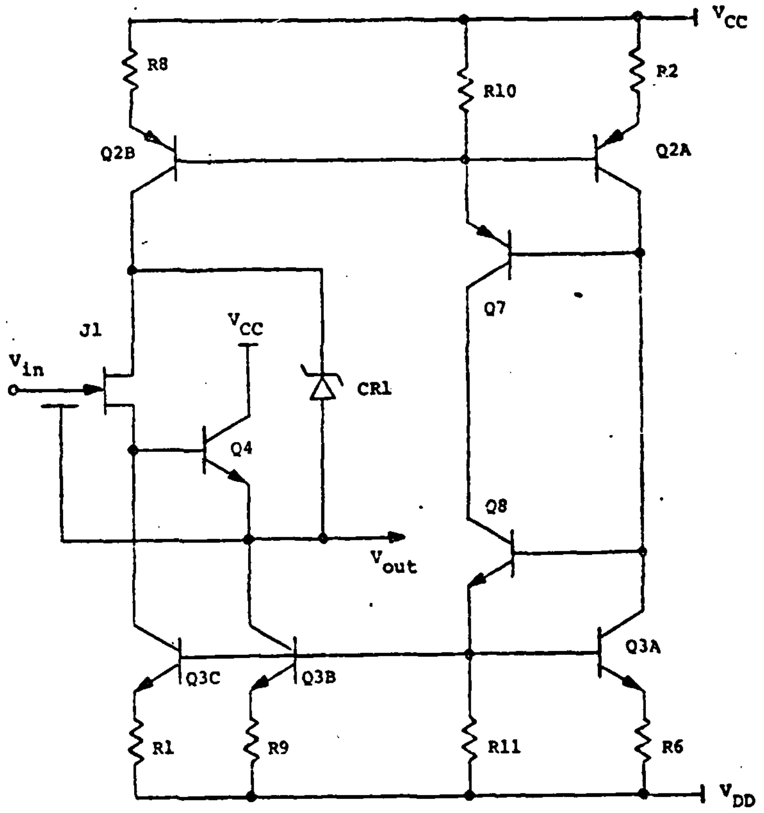


Fig. 7b: Schematic Diagram of Microprobe JJ-II

seen respectively by source transistor Q2B as well as sink transistors Q3B and Q3C are reduced - relative to figure 7a by means of emitter-follower transistors Q7 and Q8. SPICE analysis of this circuit - with transistors whose capacitance ($C_{BE}$, $C_{BC}$ and $C_{BCASE}$) are 1pF each -suggest a potential voltage bandwidth of 400 MHz with a low-frequency input - capacitance of 0.02 pF.

The bandwidth is crucially important. To develop an intuitive understanding of the problem consider a fixed-delay-system (figure 8a) and its response to a voltage-step input. Since unity gain is assumed (K = 1.0), the net charge transfer is zero. Nevertheless, substantial current-flow is induced in one direction by the input switch and later in the reverse direction by the delay switch. A plot of the SPICE run confirms (figure 8b) these speculations very nicely.

Figure 9 shows bit-line events on a U of M RAM, recorded by means of a probe built to schematic diagram 7a with provisions for case-to-gate-capacitance neutralization per 7b. The probe works. We can observe bit-line events by means of relatively inexpensive instrumentation. As expected, switching noise is conspicuous. The glitches which follow the precharge pulses are certainly objectionable. Work on this topic continues and Ms. Jiang Jialin is getting ready to write a major article on Ultrafast-Microprobes.

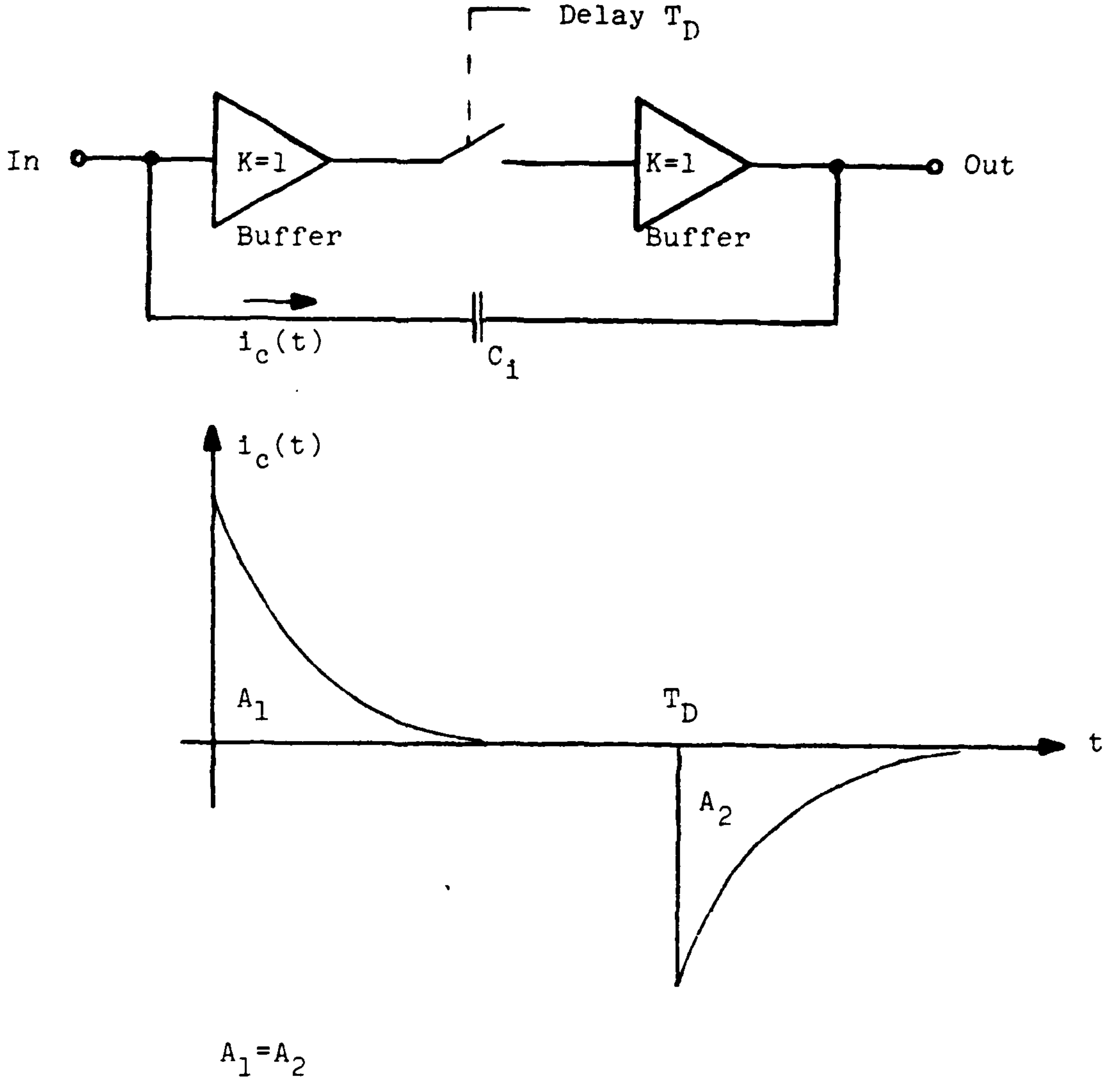
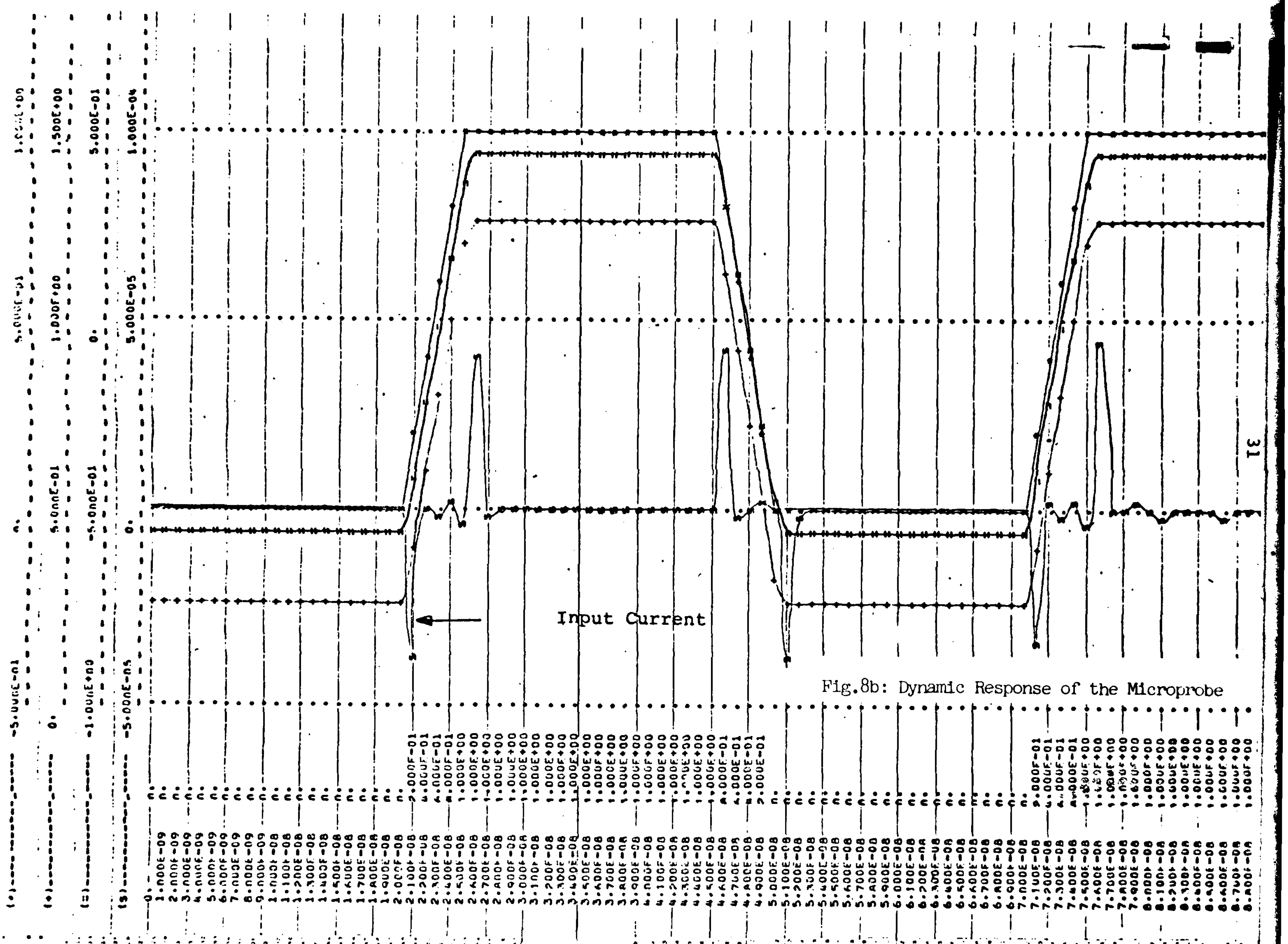
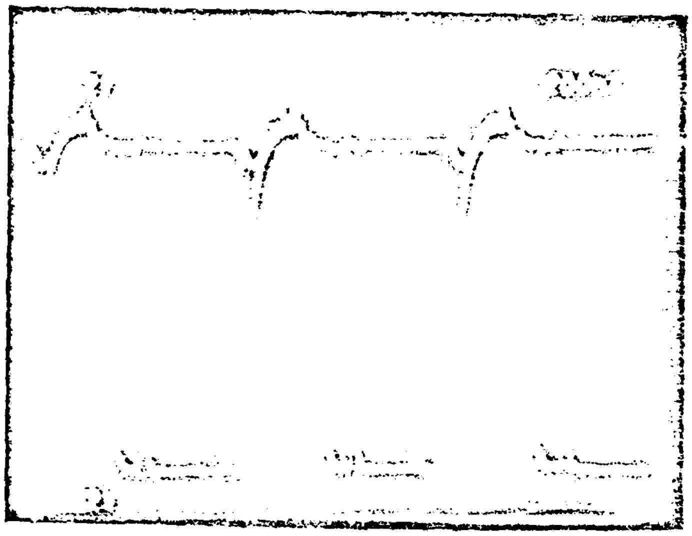


Fig.8a: Input Current in a Follower System with Discrete Delay



```
(*)------------  -5.000E-01     0.            5.000E-01     1.000E+00
(+)------------  0.             5.000E-01     1.000E+00     1.500E+00
(=)------------  -1.000E+00     -5.000E-01    0.            5.000E-01
(S)------------  -5.000E-05     0.            5.000E-05     1.000E-04
0.          0.
1.000E-09   0.
2.000E-09   0.
3.000E-09   0.
4.000E-09   0.
5.000E-09   0.
6.000E-09   0.
7.000E-09   0.
8.000E-09   0.
9.000E-09   0.
1.000E-08   0.
1.100E-08   0.
1.200E-08   0.
1.300E-08   0.
1.400E-08   0.
1.500E-08   0.
1.600E-08   0.
1.700E-08   0.
1.800E-08   0.
1.900E-08   0.
2.000E-08   0.
2.100E-08   2.000E-01
2.200E-08   4.000E-01
2.300E-08   6.000E-01
2.400E-08   8.000E-01
2.500E-08   1.000E+00
2.600E-08   1.000E+00
2.700E-08   1.000E+00
2.800E-08   1.000E+00
2.900E-08   1.000E+00
3.000E-08   1.000E+00
3.100E-08   1.000E+00
3.200E-08   1.000E+00
3.300E-08   1.000E+00
3.400E-08   1.000E+00
3.500E-08   1.000E+00
3.600E-08   1.000E+00
3.700E-08   1.000E+00
3.800E-08   1.000E+00
3.900E-08   1.000E+00
4.000E-08   1.000E+00
4.100E-08   1.000E+00
4.200E-08   1.000E+00
4.300E-08   1.000E+00
4.400E-08   1.000E+00
4.500E-08   1.000E+00
4.600E-08   8.000E-01
4.700E-08   6.000E-01
4.800E-08   4.000E-01
4.900E-08   2.000E-01
5.000E-08   0.
5.100E-08   0.
5.200E-08   0.
5.300E-08   0.
5.400E-08   0.
5.500E-08   0.
5.600E-08   0.
5.700E-08   0.
5.800E-08   0.
5.900E-08   0.
6.000E-08   0.
6.100E-08   0.
6.200E-08   0.
6.300E-08   0.
6.400E-08   0.
6.500E-08   0.
6.600E-08   0.
6.700E-08   0.
6.800E-08   0.
6.900E-08   0.
7.000E-08   0.
7.100E-08   2.000E-01
7.200E-08   4.000E-01
7.300E-08   6.000E-01
7.400E-08   8.000E-01
7.500E-08   1.000E+00
7.600E-08   1.000E+00
7.700E-08   1.000E+00
7.800E-08   1.000E+00
7.900E-08   1.000E+00
8.000E-08   1.000E+00
8.100E-08   1.000E+00
8.200E-08   1.000E+00
8.300E-08   1.000E+00
8.400E-08   1.000E+00
8.500E-08   1.000E+00
8.600E-08   1.000E+00
8.700E-08   1.000E+00
8.800E-08   1.000E+00
```

Fig.8b: Dynamic Response of the Microprobe



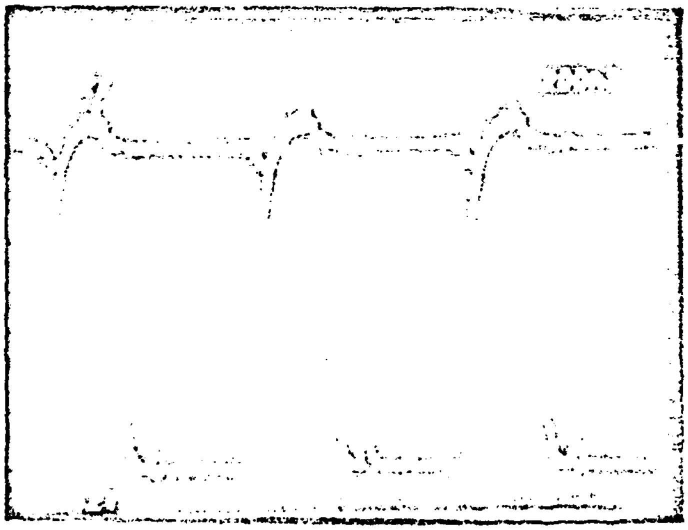


Fig. 9: Recording of BIT-LINE Events

## 4.2 Pseudo-Analog-Testing (PAT)

The technique of Pseudo Analog Testing has been conceived and developed by us to meet the pressing demand for a tool which would truly perform the functions of an Ultra-Fast Zero-Input-Capacitance Voltage-Follower. It emerged in the course of Gedenken Experimentation with Analog Testing of Digital Devices, another idea formulated by us. Details of PAT are given in appendix 4 and reference 30. Here we merely point out that the concept of Pseudo Analog Testing can be put to work in many ways, some of them non-destructive in nature. We have, for example, monitored the "best-case" refresh time of individual cells to obtain absolute and relative data on the leakage currents of storage capacitors. There is also dynamic PAT(31), the original form of which - proposed by us in reference 19 (appendix 3) - has been popularized by Teradyne under the label of "Bump, Bounce and Rebound Testing" (32).

Pseudo Analog Testing is limited in scope, but, in certain cases, it yields credible quantitative results where other methods fail. No specialized equipment is required; in contrast to Stroboscopic Scanning Electron Spectroscopy available only at the likes of IBM and Bell Telephone, Pseudo Analog Testing is within the reach of everybody. Used as in section 5.1, it is destructive but the loss of a chip is the least of our concerns when VLSI diagnostics are attempted.

## 5.0 The U of M 16k-RAM

The U of M 16k-RAM (figure 10) was designed and processed in cooperation with the Microcircuit Division of the Control Data Corporation. This state-of-the-art Oxide Isolation NMOS chip facilitates MICROPROBE as well as PAT observation of circuit peculiarities and process statistics.

There are the usual 128 rows and 128 columns but the reference capacitors come in 8 different sizes. Twenty-micron-square Aluminum-Pads over Field-Oxide attach to the individual section of bit lines through five-micron-square contact windows. Provisions are also made for easy connection of PAT poteniometers to the precharge transistors (nodes $X_s$, $X_L$ and $X_R$ in figure 3).

The U of M chip is available, for qualifying research work, from Mr. Charles T. Naber, Research Manager, Control Data Corporation, Minneapolis, Minnesota 55440.

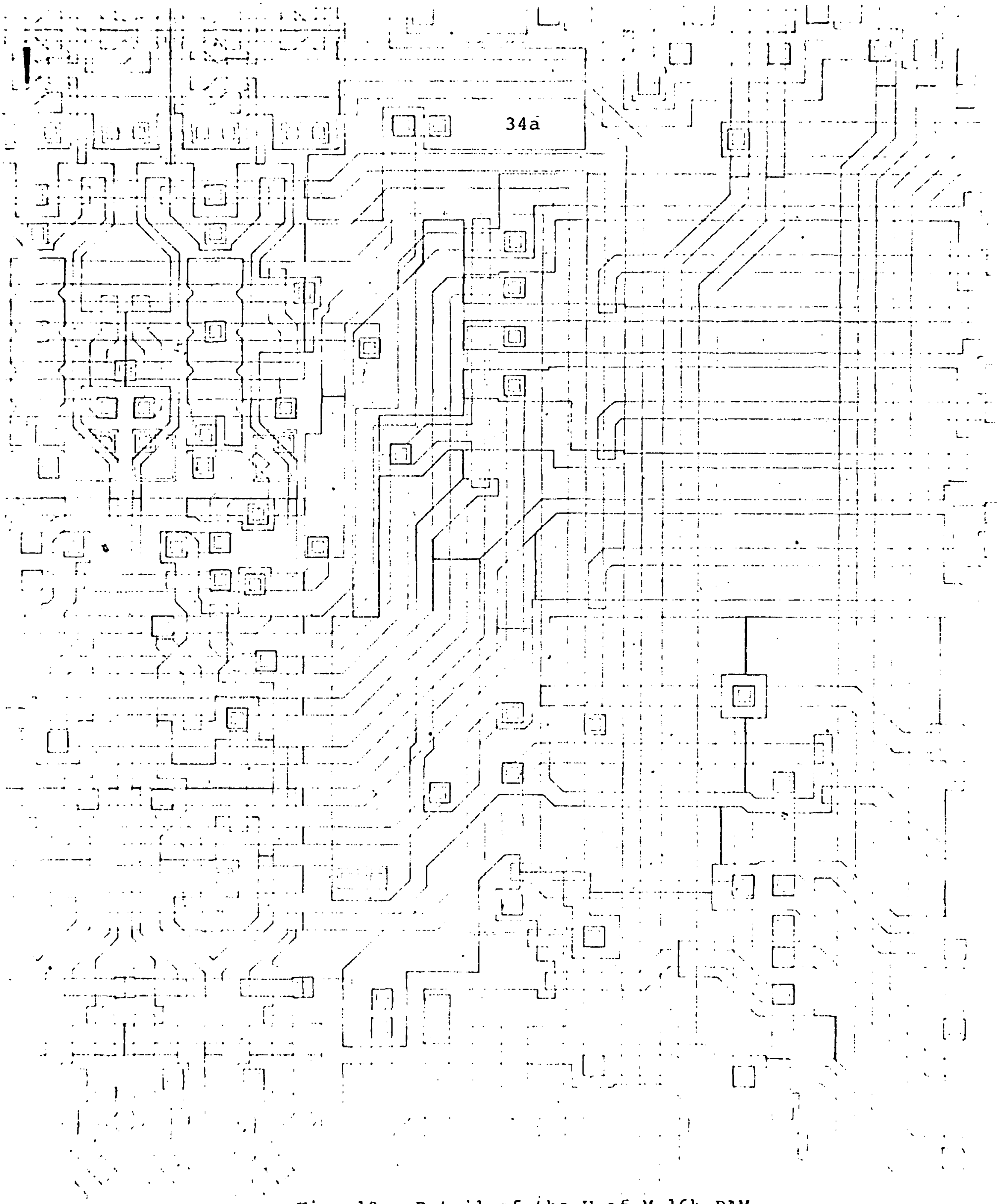


Fig. 10: Detail of the U of M 16k-RAM

## 5.1 The Fault Profile of the U of M RAM

The logic diagram of the Raster-Scan Memory-Exerciser developed by us for Pseudo-Analog-Testing is given in figure 11. Though the scanning system always works its way through all 16,384 location, the display can be set to show either the entire chip or just one quadrant. The optical resolution of the two settings are displayed in photo 1 of figure 12: individual bits are clearly discernible in the single quadrant pictures; full-chip images give a good presentation of row and column events. For another preliminary observation turn to photos 12.5 and 12.6. Chip number 5 was shot twice, once at 5 p.m. and again at 7 p.m. of 5/17/81. The two pictures are identical, as far as one can tell, demonstrating that Pseudo Analog Testing does deliver reproducible results.

The devices whose performance is displayed in photos 12.1 through 12.13 are regular production chips (not U of M RAMs). Examining their behavior we look - to begin with - for average performance margins. With a HIGH voltage (nominally 12V on the top plate), errors in the form of FALSE ZEROS set in on the lower sector (left-hand side of the BIT-LINE in figure 12.10) at 40 mV with a mean of about 110 mV; complete switch-over occurs at 80 to 160 mV.

More details appear in table 1. The results are consistent on one hand and surprising on the other. The "worst case" performance margin is 40 mV, The mean is to 100 mV in the set which comprises chips number 1,3,5,6,7, and 11, made by manufacturer A. Chip M1, processed by vendor M has a mean of about 200 mV. Note that the worst case is established at 40 mV on chip 7 by just one

| Chip # | FALSE ZEROS | | | | FALSE ONES | | | | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| | Left Segment | | Right Segment | | Left Segment | | Right Segment | | |
| | min. | max. | min. | max. | min. | max. | min. | max. | |
| 1 | 60 | 80 | 100 | 120 | 120 | 140 | 60 | 100 | Only 2 FALSE ONES at 60 mV |
| 3 | 80 | 120 | 100 | 120 | 120 | 180 | 60 | 140 | Only 1 FALSE ONE at 60 mV |
| 5 | 80 | 120 | 60 | 80 | 80 | 120 | 120 | 140 | 1 Soft Right Segment at 60 mV |
| 6 | 60 | 140 | 60 | 120 | 80 | 200 | 80 | 160 | Pseudo-Hard Errors |
| 7 | 80 | 140 | 60 | 140 | 40 | 140 | 100 | 160 | Only 1 False ONE at 40 mV |
| 11 | 80 | 120 | 100 | 120 | 140 | 180 | 140 | 180 | |
| Average | 73 | 120 | 100 | 117 | 97 | 160 | 93 | 147 | |
| Mean | 96 | | 98 | | 128 | | 120 | | |
| M1-Mean | ~ 200 | | ~ 200 | | ~ 200 | | ~ 200 | | 1 Soft Bit Line at 144 mV |

Table 1: Fault Profile of 16k-RAMs

cell; all other cells on the same chip - all 16,383 of them - have performance margins that are at least twice as high. Again, on chip three (figure 12.4), the worst case of 60 mV is produced by just one cell (different than on chip 7) while all other cells work to and beyond 100 mV.

The wide distribution of performance margins is disturbing; it leads to soft errors and must, therefore, be corrected. In any case, why should one cell be much worse then the rest? We expect that minute "soft" (non-catastrophic) process defects - unduly thin Gate Oxide, for example - are at the bottom of this puzzle. Investigations of this phenomenon will be continued in a MASTER THESIS of James E. Broughton from SPERRY-UNIVAC in St. Paul, Minnesota.

To look at a "soft defect" in reference cells turn to photograph 12.7 (chip #6-ZEROS). The whole lower (left-hand) sector of a bit line has an unduly low performance margin. There is almost definitely a flaw in the components associated with the right-hand storage cell; a Sense-Amp offset problem would have shown up, for examples, as a bad sector on the left for FALSE ONES and an equally bad right sector of the same BIT-LINE for FALSE ZEROS, as in figures 12.12 and 12.13.

Apart from the above noted exceptions, the spread of the switch-over voltage is contained to 20 mV. Concurrently, the reproducibility evidence submitted with regard to chip #5 in photographs 12.5 and 12.6 demonstrates that the pertinent noise levle is relatively low. One can assume, therefore, that the 20 mV straggle is attributable to variations in storage capacitance. With a mean of 100 mV, we get a typical variation of about 20%. This is acceptable, but the aforementioned "worst case" is not. As already noted, the wide dispersion of capacitance values will lead to performance marginalities which precipitate soft errors. Two action items are called for:

1) Reduce the parasitic capacitance of the BIT-LINE to improve the overall performance margin, and
2) Clean-up the process to reduce cell-to-cell variations of capacitance.

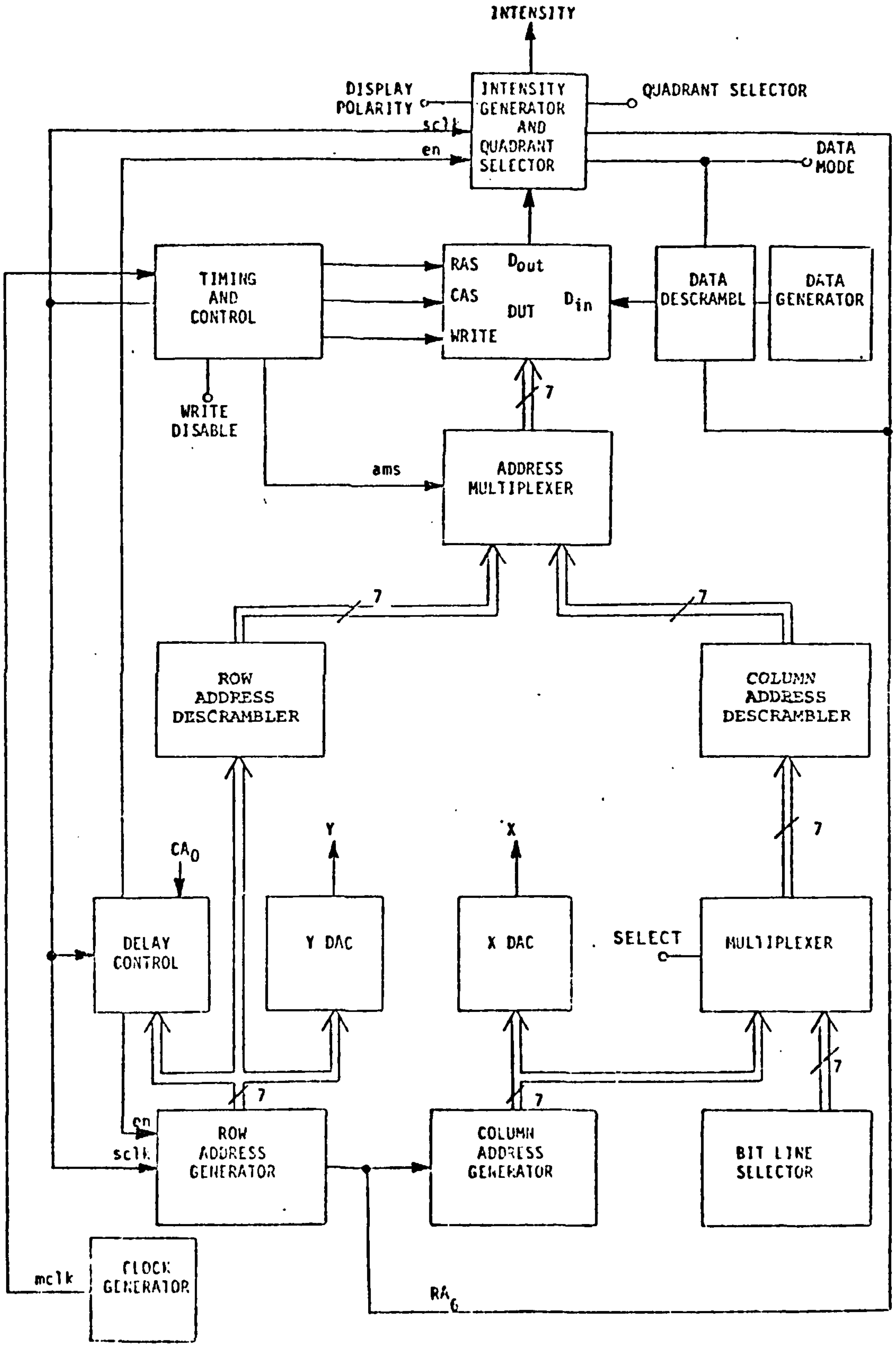
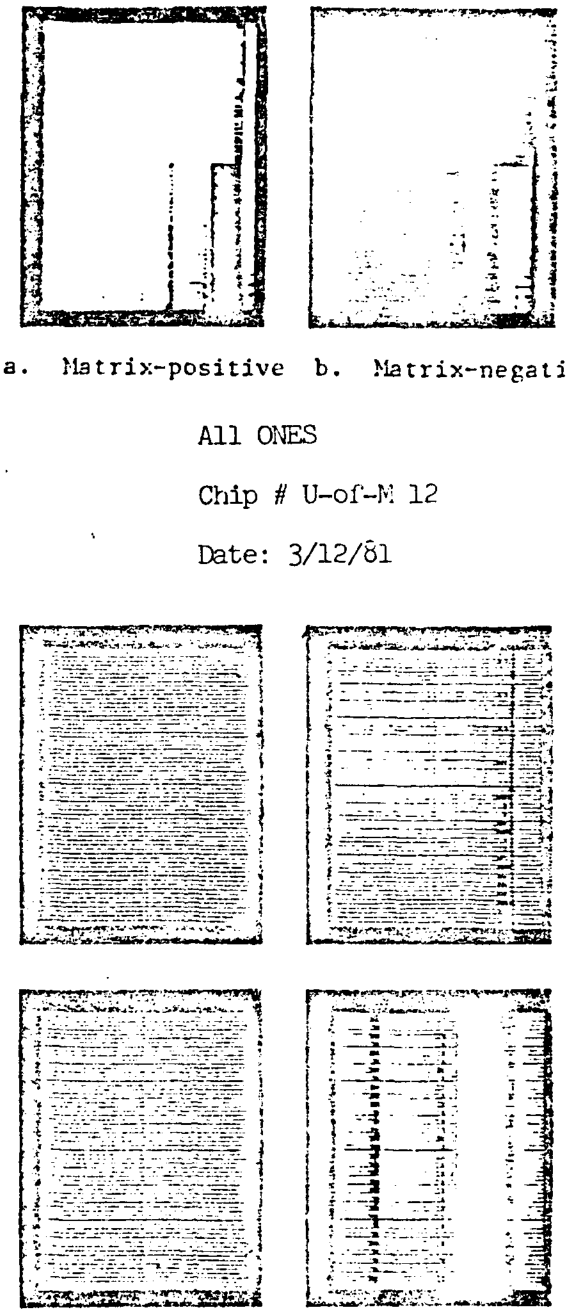


Fig. 11: Raster-Scan Memory-Exerciser



a. Matrix-positive b. Matrix-negative

All ONES

Chip # U-of-M 12

Date: 3/12/81

c. Quadrants-positive

Fig.12.1: Resolutions of the BIT-MAP display

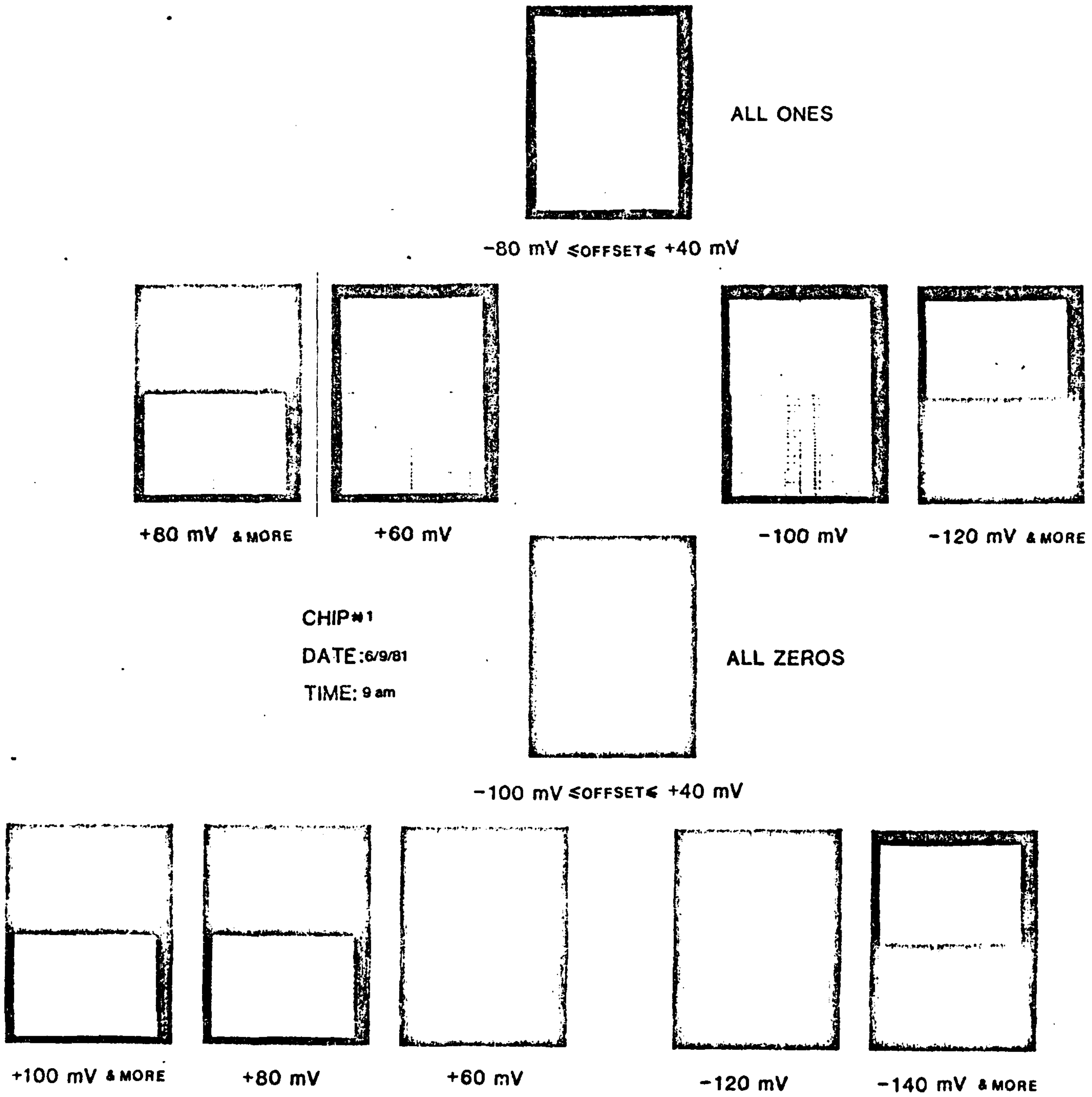
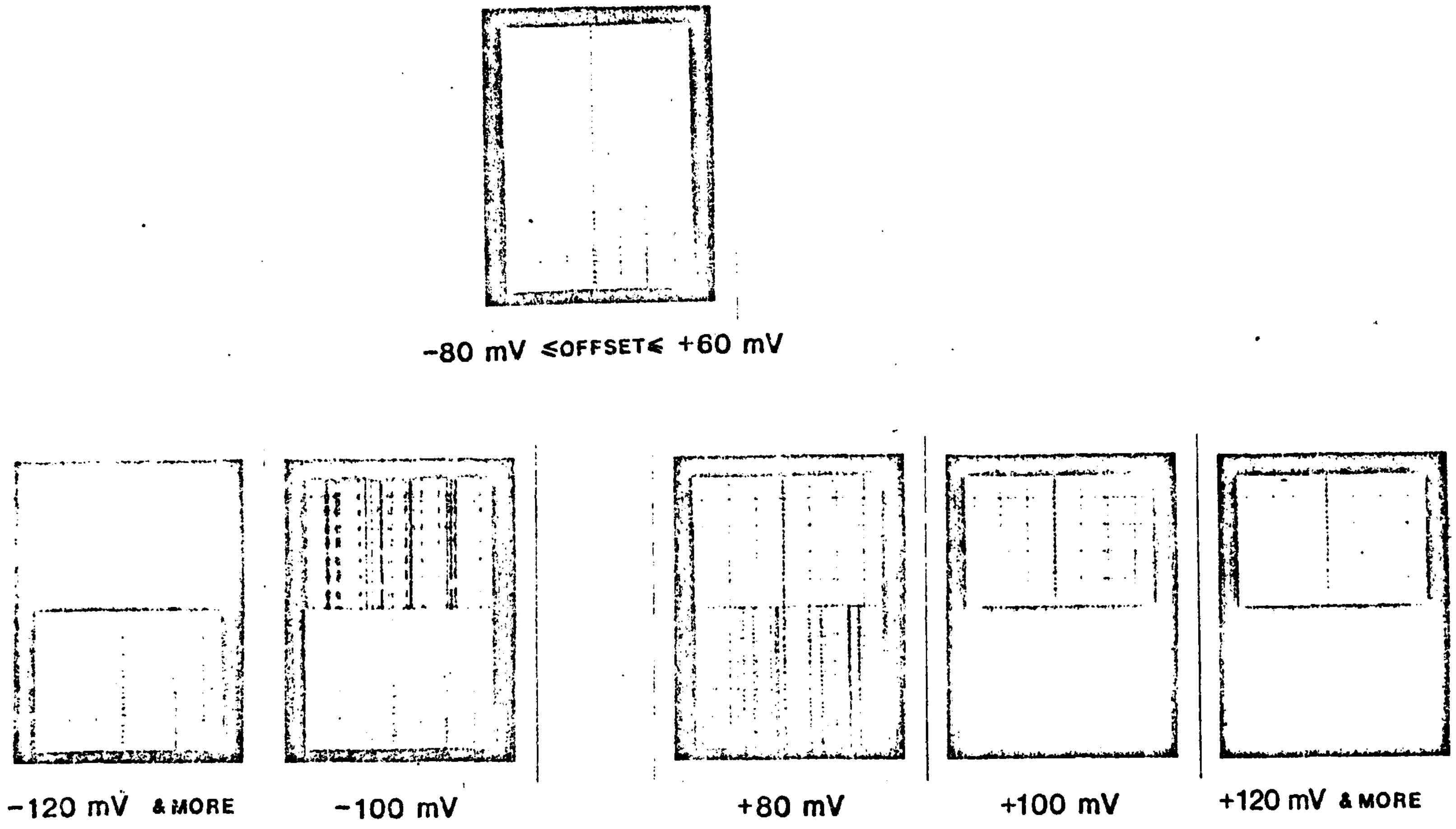

ALL ONES
-80 mV ≤OFFSET≤ +40 mV
+80 mV & MORE
+60 mV
-100 mV
-120 mV & MORE
CHIP#1
DATE:6/9/81
TIME: 9 am
ALL ZEROS
-100 mV ≤OFFSET≤ +40 mV
+100 mV & MORE
+80 mV
+60 mV
-120 mV
-140 mV & MORE

Fig.12.2



ALL ONES
CHIP # 3
DATE: 6/12/81
TIME: 3 pm

Fig.12.3

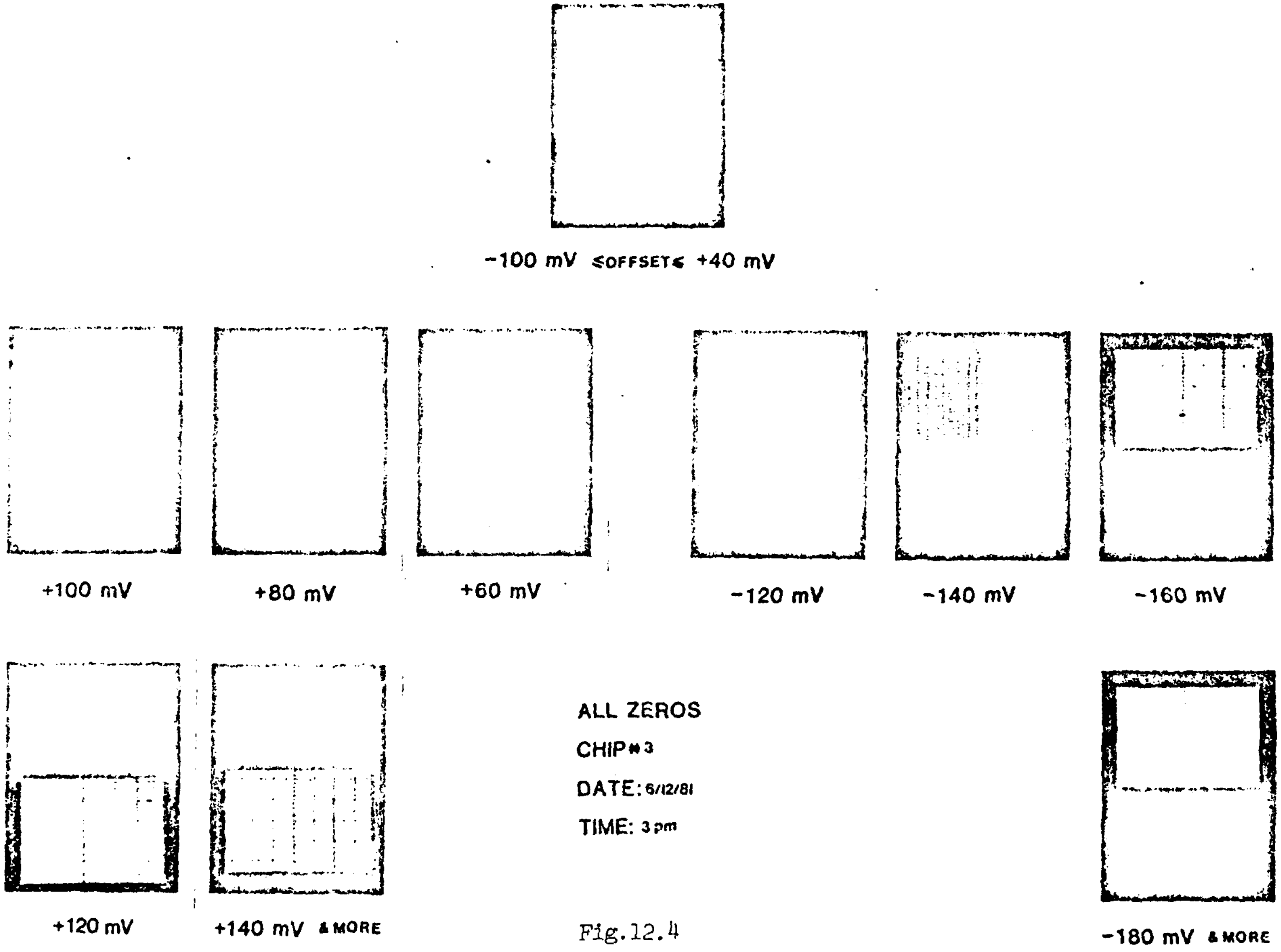

-100 mV ≤OFFSET≤ +40 mV
+100 mV
+80 mV
+60 mV
-120 mV
-140 mV
-160 mV
+120 mV
+140 mV & MORE
ALL ZEROS
CHIP#3
DATE: 6/12/81
TIME: 3 pm
-180 mV & MORE

Fig.12.4

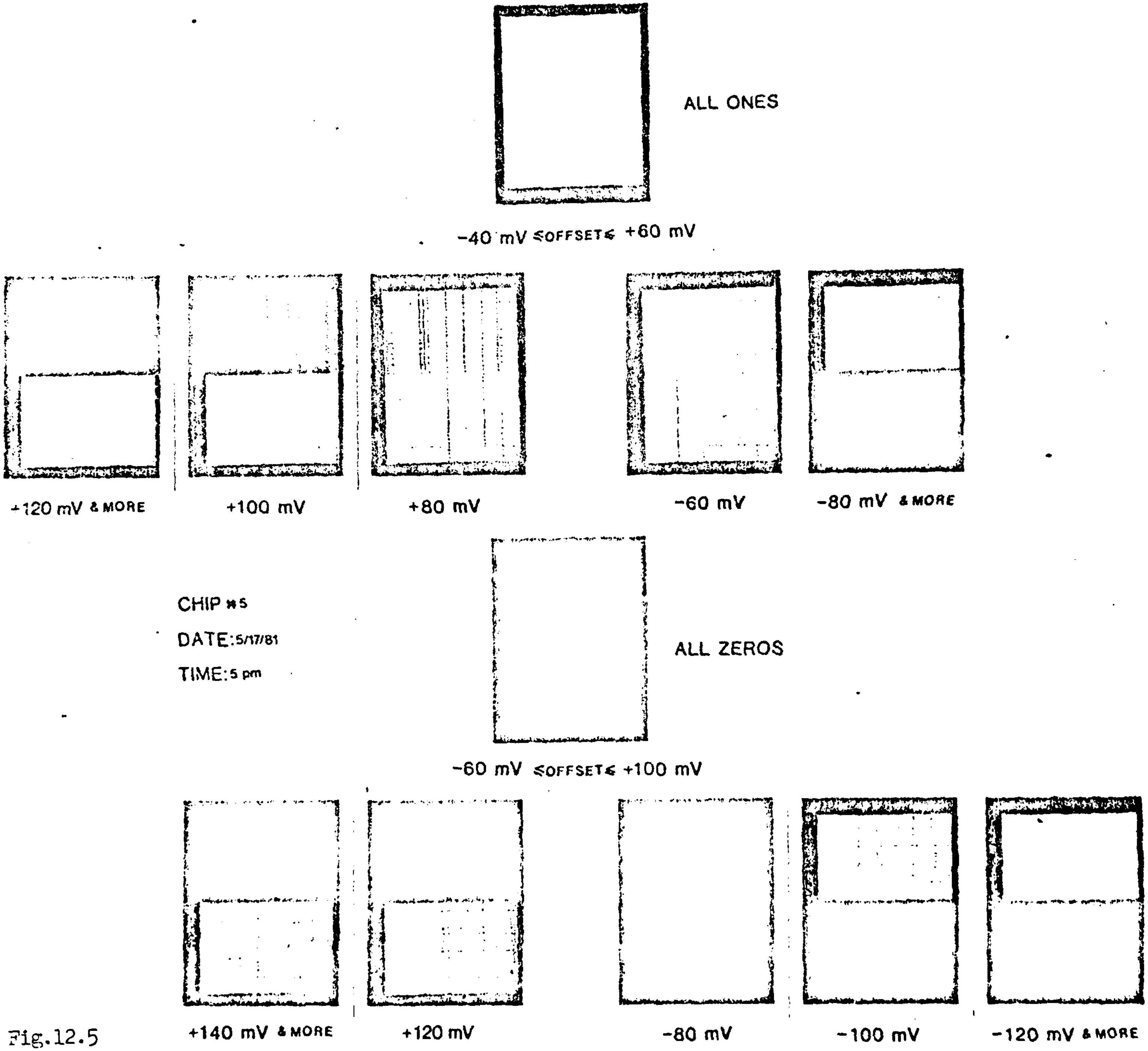

ALL ONES
-40 mV ≤OFFSET≤ +60 mV
+120 mV & MORE
+100 mV
+80 mV
-60 mV
-80 mV & MORE
CHIP #5
DATE:5/17/81
TIME:5 pm
ALL ZEROS
-60 mV ≤OFFSET≤ +100 mV
+140 mV & MORE
+120 mV
-80 mV
-100 mV
-120 mV & MORE

Fig.12.5

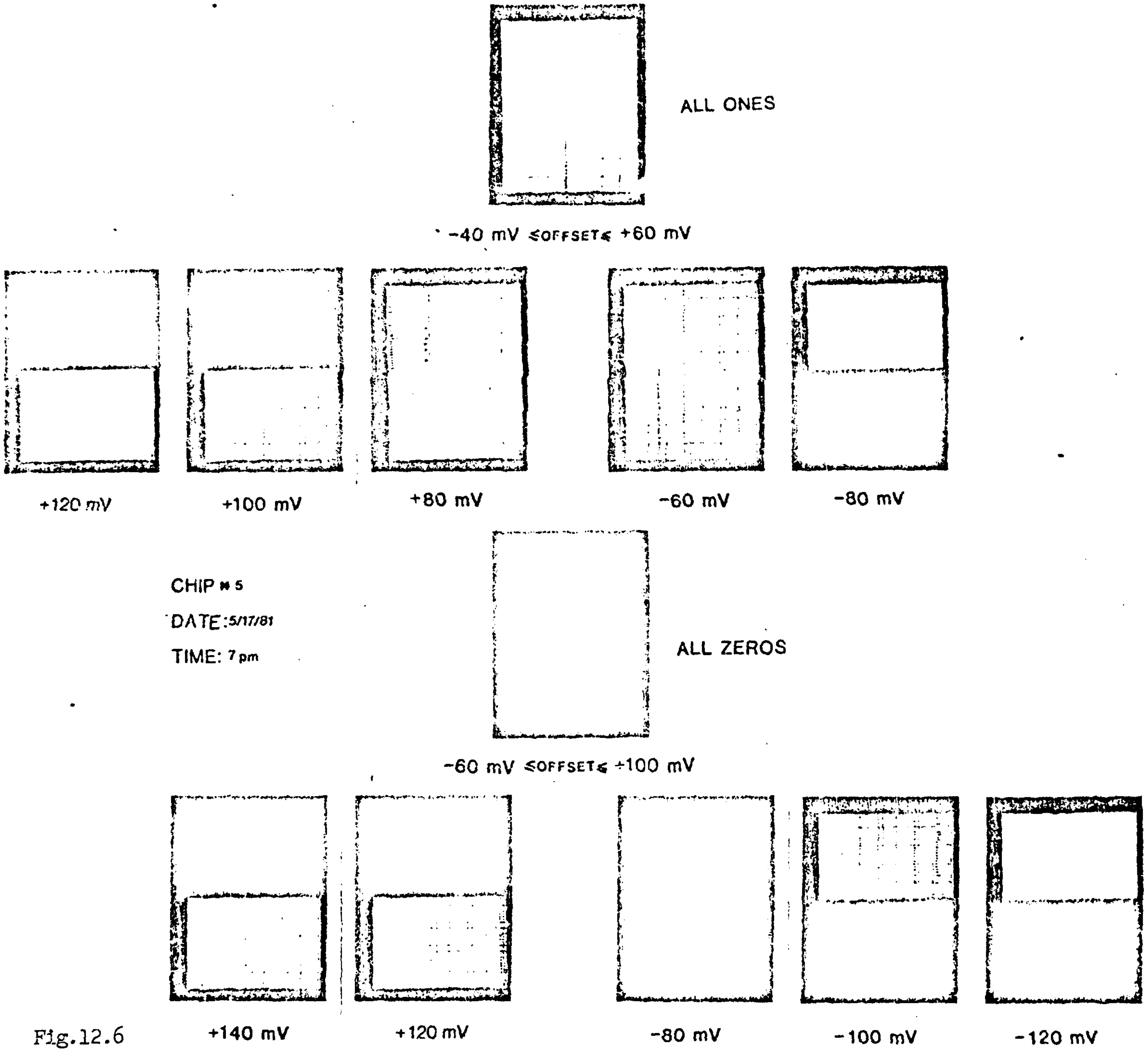

ALL ONES
-40 mV ≤OFFSET≤ +60 mV
+120 mV
+100 mV
+80 mV
-60 mV
-80 mV
CHIP # 5
DATE:5/17/81
TIME: 7 pm
ALL ZEROS
-60 mV ≤OFFSET≤ +100 mV
+140 mV
+120 mV
-80 mV
-100 mV
-120 mV

Fig.12.6

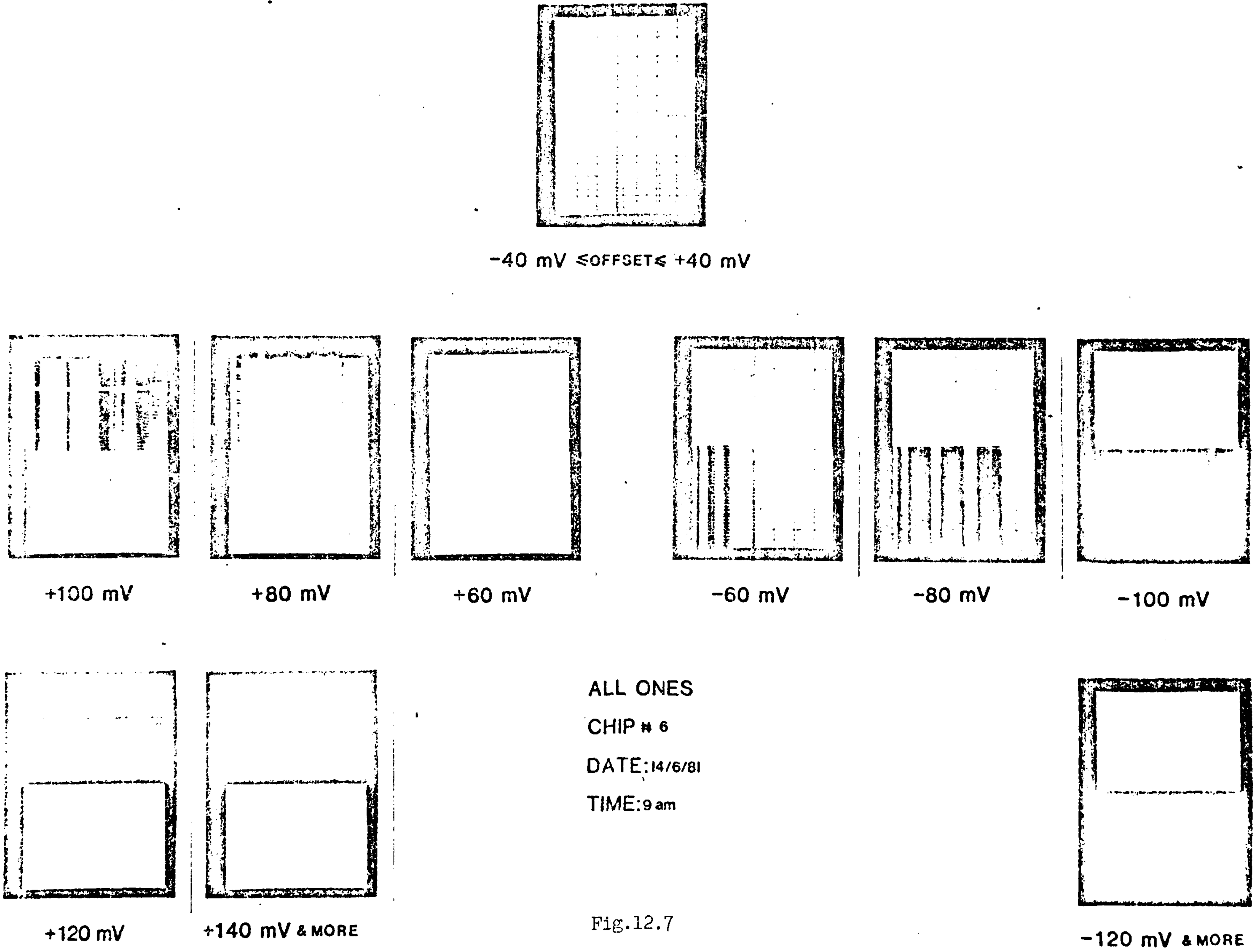

-40 mV ≤OFFSET≤ +40 mV
+100 mV
+80 mV
+60 mV
-60 mV
-80 mV
-100 mV
+120 mV
+140 mV & MORE
ALL ONES
CHIP # 6
DATE:14/6/81
TIME:9 am
-120 mV & MORE

Fig.12.7

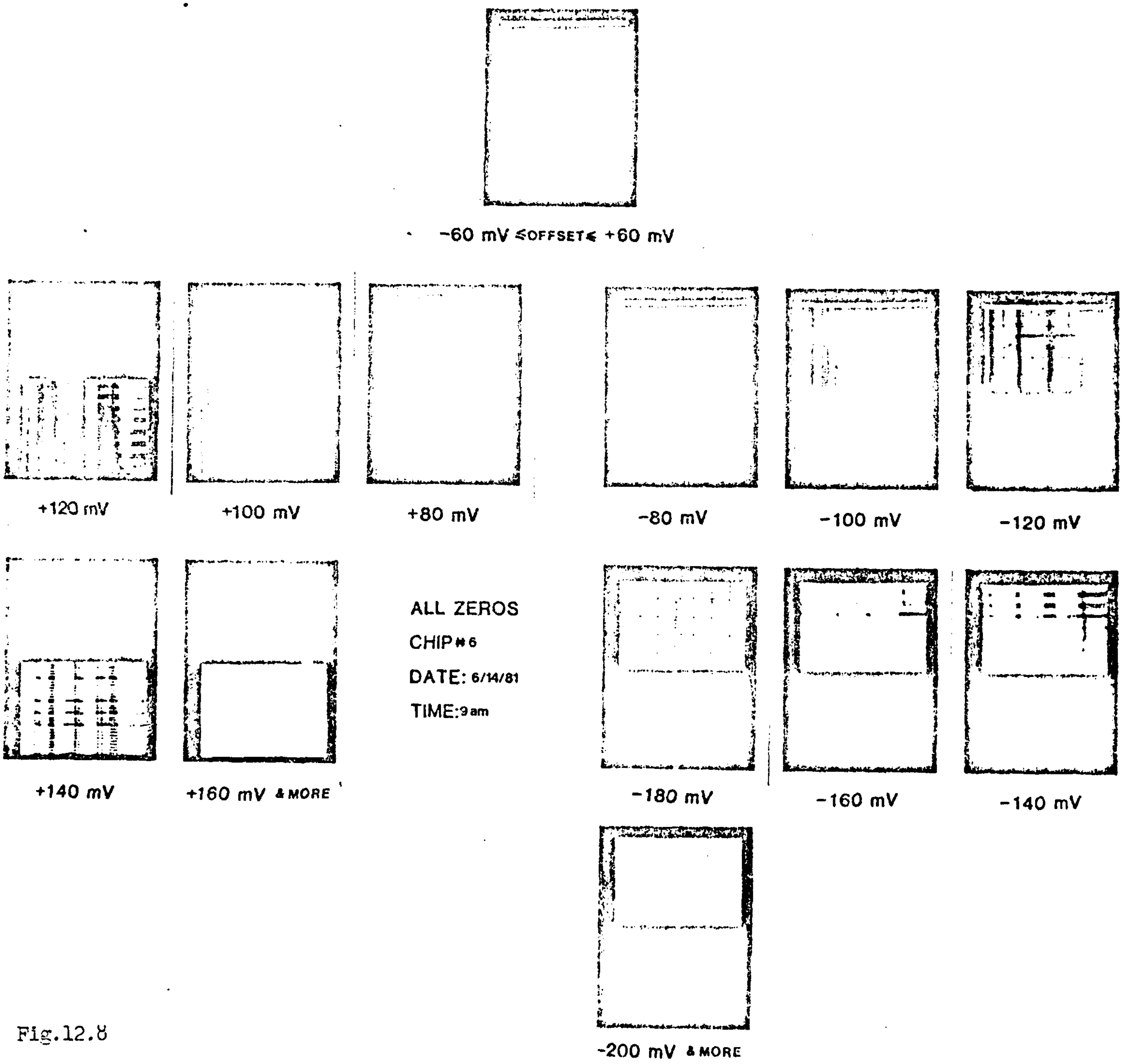

-60 mV ≤OFFSET≤ +60 mV
+120 mV
+100 mV
+80 mV
-80 mV
-100 mV
-120 mV
+140 mV
+160 mV & MORE
ALL ZEROS
CHIP#6
DATE: 6/14/81
TIME:9am
-180 mV
-160 mV
-140 mV
-200 mV & MORE

Fig.12.8

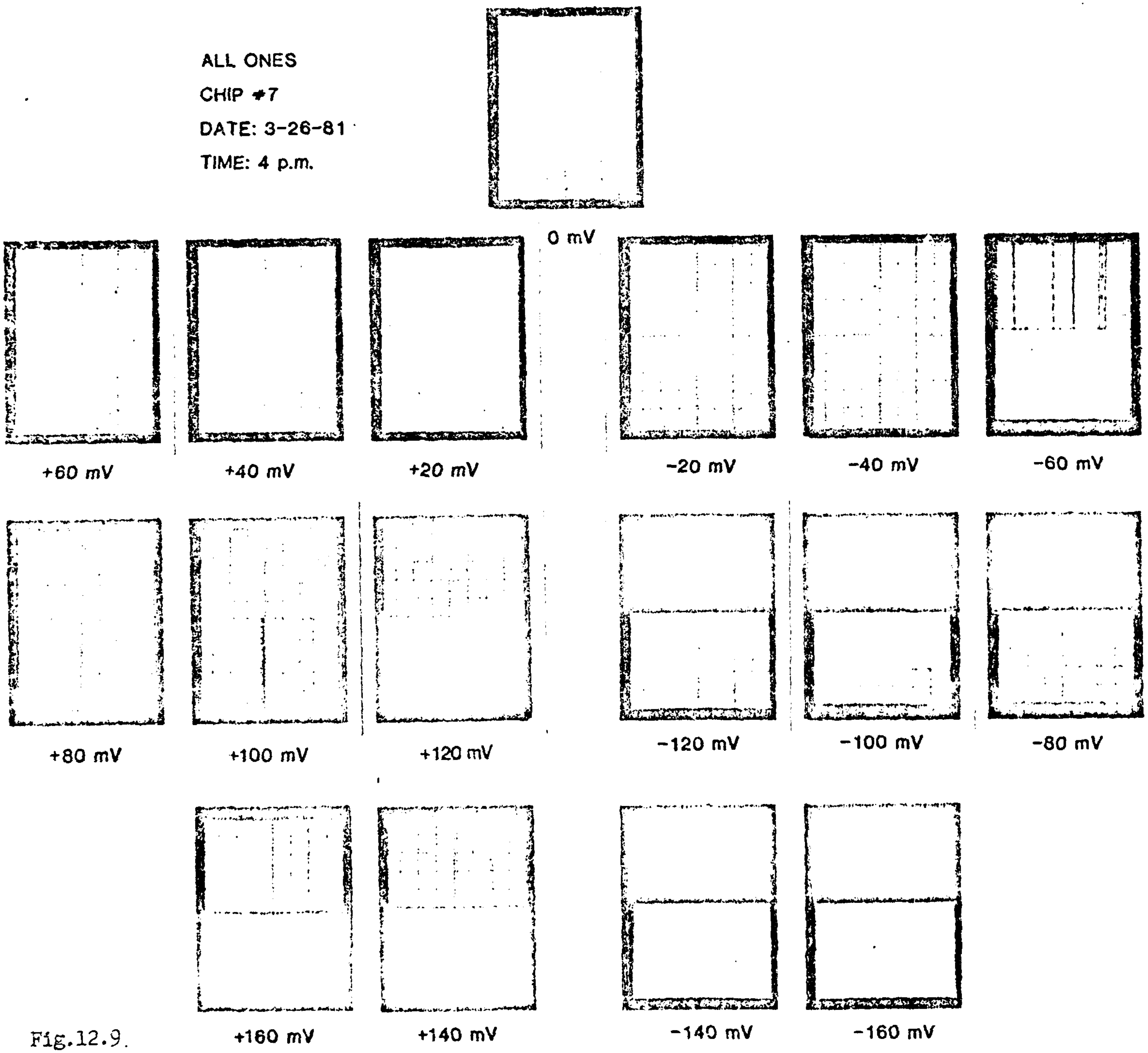

ALL ONES
CHIP #7
DATE: 3-26-81
TIME: 4 p.m.
0 mV
+60 mV
+40 mV
+20 mV
-20 mV
-40 mV
-60 mV
+80 mV
+100 mV
+120 mV
-120 mV
-100 mV
-80 mV
+160 mV
+140 mV
-140 mV
-160 mV

Fig.12.9.

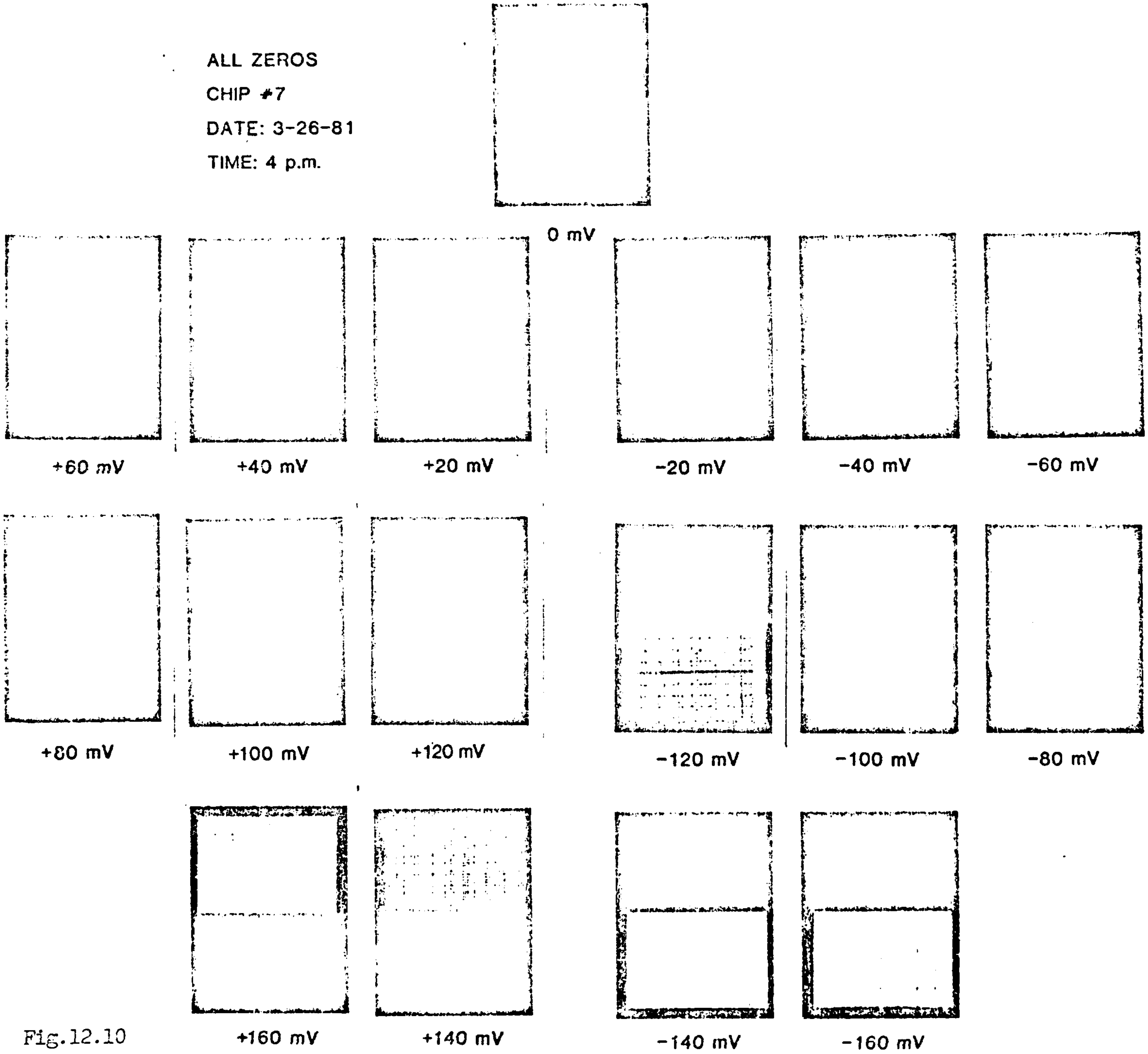

ALL ZEROS
CHIP #7
DATE: 3-26-81
TIME: 4 p.m.
0 mV
+60 mV
+40 mV
+20 mV
-20 mV
-40 mV
-60 mV
+80 mV
+100 mV
+120 mV
-120 mV
-100 mV
-80 mV
+160 mV
+140 mV
-140 mV
-160 mV

Fig.12.10

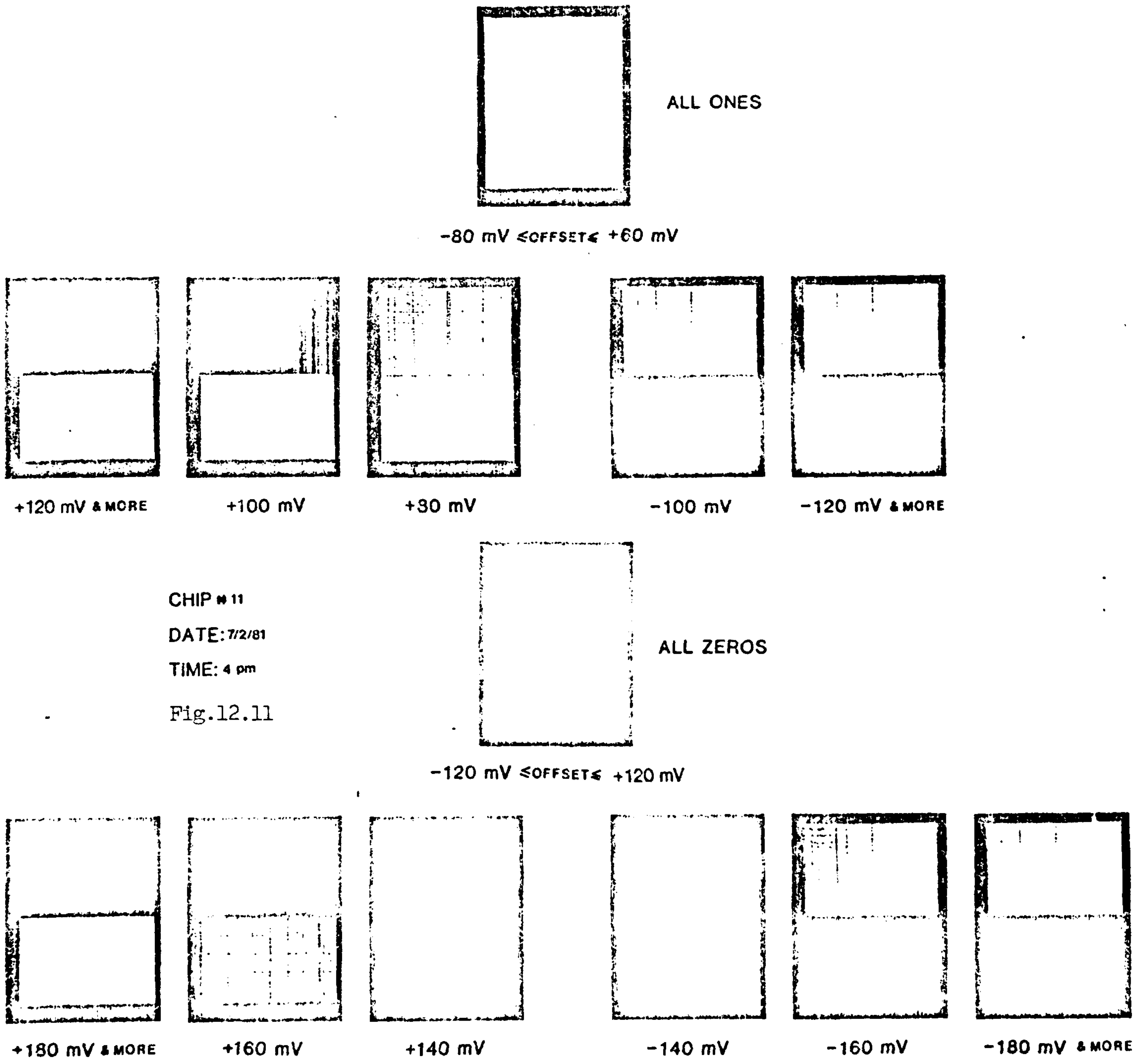
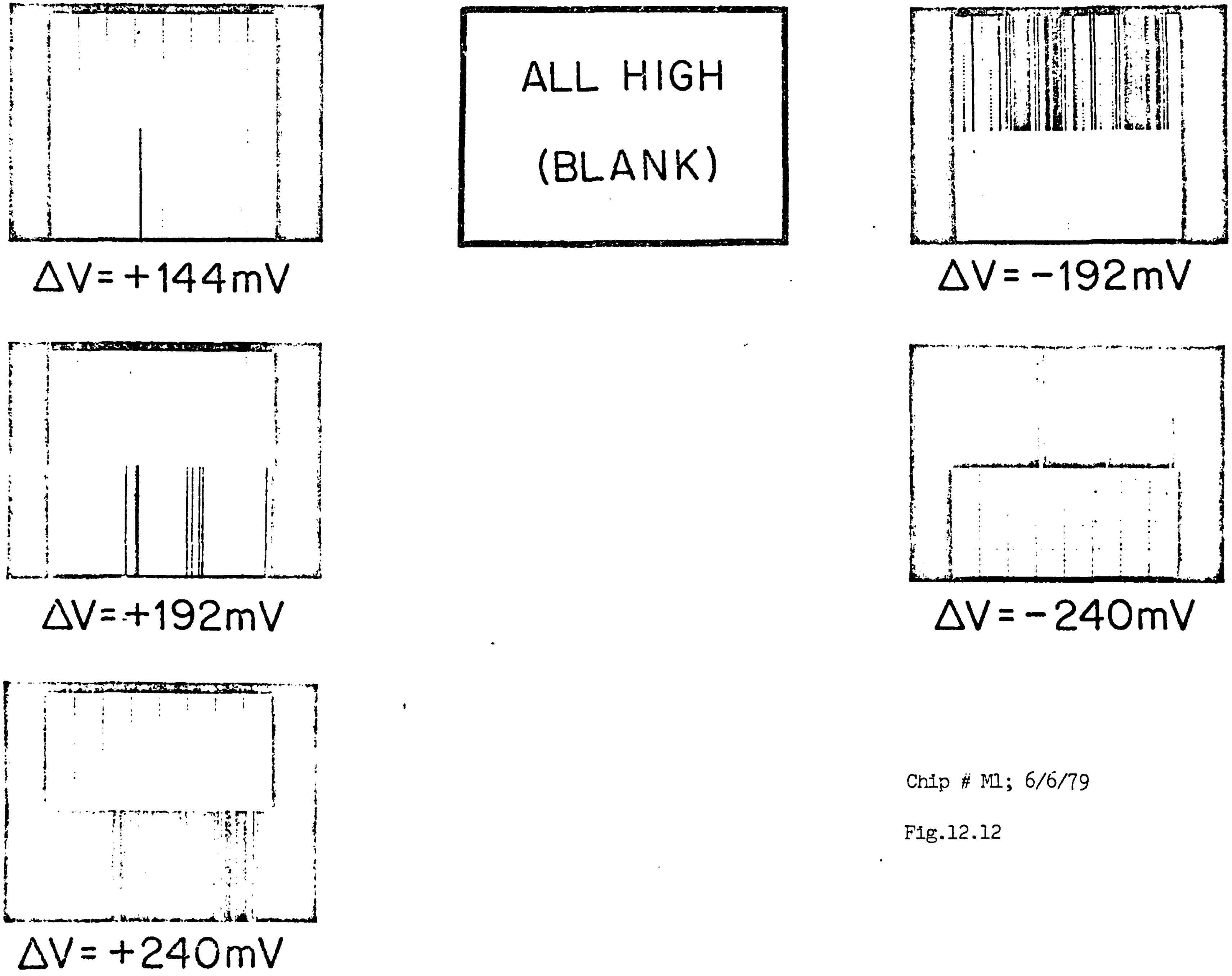

ALL ONES
-80 mV ≤OFFSET≤ +60 mV
+120 mV & MORE
+100 mV
+30 mV
-100 mV
-120 mV & MORE
CHIP # 11
DATE: 7/2/81
TIME: 4 pm
ALL ZEROS
-120 mV ≤OFFSET≤ +120 mV
+180 mV & MORE
+160 mV
+140 mV
-140 mV
-160 mV
-180 mV & MORE

Fig.12.11



ΔV=+144mV

ΔV=+192mV

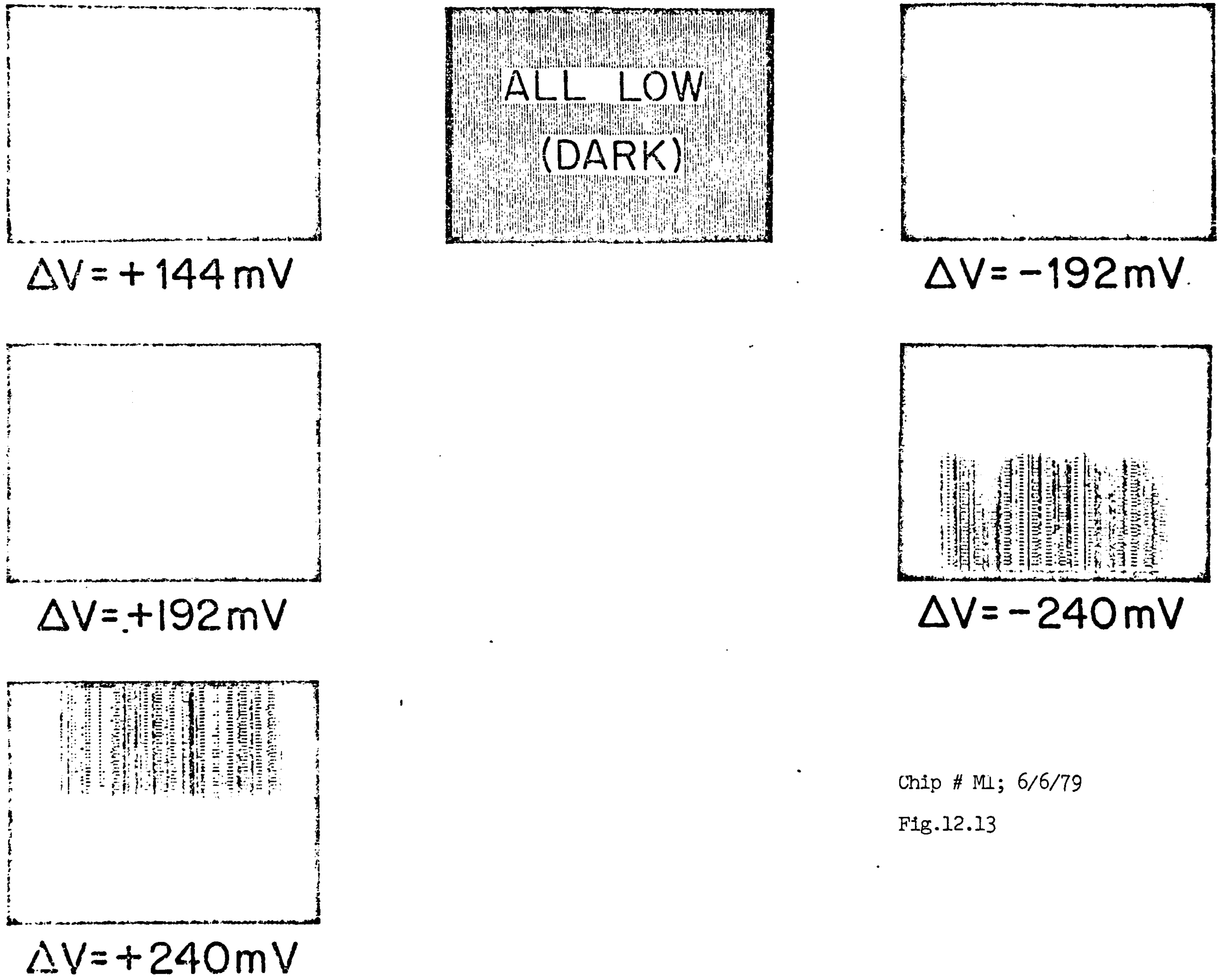
ΔV=+240mV

ΔV=-192mV

ΔV=-240mV

Chip # M1; 6/6/79

Fig.12.12



Chip # M1; 6/6/79

Fig.12.13

## 6.0 Reliable Chips and Systems

System reliability begins at the "materials" level. In contrast to past practice, we want epi-wafers, rather than their homogeneous counterparts, in order to immunize our products as far as possible against malefactions of alpha particles [33]. We also want a low dislocation density to assure low leakage, but most important of all is uniformity, since poor uniformity invariably leads to Pattern Sensitivity, the deathknell of reliability.

Processing must be uniform too: cell to cell, chip to chip, wafer to wafer and run to run. Statistical straggle reduces performance margins and thus leads to hard faults or, worse still, to soft errors. The matter of yield is significant. Low yield implies poorly centered processing, or excessive variations or inadequate performance margins. Either one of these will result in low yield as well as high Pattern and Sequence Sensitivity.

In circuit design and chip layout, most dangerous are minor flaws. Routine runs of CAD programs will never catch them. One needs to develop the technique of adversary analysis - a technique aimed at the identification of worst case combinations - and one also needs to establish a proper data base for whatever process one happens to work with.

In architecture, crucial is the reduction of the risk of burst errors. The chip as well as the entire system must be designed with this thought in mind. One already operates on multiple operands for the sake of low overheads [appendix 2] but one should go a step further and consider block-organization of memories for the explicit purpose of effective error correction.

Altogether one needs good epi, uniform processing, wide performance margins, effective error correction and penetrating diagnostics. More work must be done in the area of process satistics; figures 12.1 through 12.13 demonstrate that very eloquently. Also needed are means for the quantization of process integrity and design quality. Last but not least, there is a call for diagnostic tools such as the Scanning Stroboscopic Electron Microscope and our own Microprobe or, as long as nothing better is available, the Pseudo Analog Test Technique.

## 7.0 Conclusion

Testing should not be treated as an objective in its own right. Verification of performance, rather than testing, is the real objective. It won't do to argue that we must test to be "certain"; there is no certainty, not ever. We must confine ourselves to statistical expectations, asking for appropriately high confidence levels and no more. In VLSI one runs exhaustive general purpose tests for hard errors and a few highly customized tests for non-random soft errors. The latter operation is far from being exhaustive. One deals with crosstalk in the form of Pattern and Sequence Sensitivity. Anything in excess of 30 bits (sic) leads to absurd situations.

To curb Pattern and Sequence Sensitivity, we need KNOWLEDGE above all; knowledge of circuitry as well as processing and not just superficial knowledge at that. Pattern and Sequence Sensitivities are a matter of marginalities; to eliminate them, we must understand the subtleties of circuitry and processing in order to identify worst case situations precipitated by theoretical circuit problems on one hand and the very real process peculiarities on the other.

The need for a data base complete with statistical information will become even more pressing as we progress from the present five micron features to submicron dimensions. Reproducibility of characteristics does depend distinctly on feature size; we may want to trade size against matching. There exist good reasons for fairly large Sense Amplifiers in otherwise submicron memories.

## 8.0 References

1. A. Tuszynski, "Self-Learning Machines Applied to the Testing of Semiconductor Memories", Autotestcon 78, pp. 246-250.

2. E. B. Eichelberger and T. K. Williams, "A Logic Design Structure for LSI Testability", 14th Design Automation Conference, 1977, pp. 462-468.

3. A. Tuszynski, "Skirting Thin-Film Design Problems", Microelectronics Design, H. Bierman - Editor, Hayden Co., 1966, pp. 79-85.

4. Editorial staff, "A New Perspective on IBM Technology: Semiconductors", IBM, 1980.

5. S. John De Falco, "Predicting Crosstalk in Digital System", Computer Design, June 1973, pp. 69-75.

6. Chang, Manning & Metze, "Fault Diagnosis of Digital Systems", John Wiley, 1970.

7. Friedman & Menon, "Fault Detection in Digital Circuits", Prentice Hall, 1971.

8. Breuer & Friedman, "Diagnonsis & Reliable Design of Digital Systems", Computer Science Press, 1976.

9. Paul Roth, "Computerlogic, Testing and Verification", Computer Science Press, 1980.

10. Joint Electron Devices Engineering Council (JEDEC), publication #JC-42-78-65.

11. Pickel and Blandford, "Cosmic Ray Induced Errors in MOS Memory Cells", 1978 IEEE Conference on Nuclear and Space Radiation Effects, paper #X78-3171501.

12. Alan Lewandowski, "Effect of Alpha Particles on Dense Storage Devices", MS Thesis, U of M, 1979.

13. May and Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories", IEEE Trans. on Electron Devices, Jan. 1979, pp. 2-9.

14. Editorial, "Polyimide Prevents Alpha Errors", Electronic Design, 1 September 1980, p. 32.

15. John Gosch, "Polymerdoubles as photoresist and insulator", Electronics, 16 June 1981, pp. 73-74.

16. E. J. Riley Jr., "Determining Trace Uranium in Ceramic Memory Packages Using Neutron Activation with Fission Track Counting", Semiconductor International, May 1981.

17. R. T. Chien, "Memory Error Control: Beyond Parity", IEEE Spectrum, July 1973, pp. 18-23.

18. A. Tuszynski, "Medium and Soft Errors", 1980 IEEE Workshop on Memory Testing, pp. 1-16.

19. Rinerson and Tuszynski, "Identification of Causes of Pattern Sensitivity", IEEE LSI Test Symposium, 1977, pp. 246-250.

20. Kuo, Kitagawe, Klard & Drayer, "Sense Amplifier Design is Key to 1-transistor Cell in a 4096 RAM", Electronics, 13 September 1973, pp. 116-121.

21. C. N. Ahlquist et al., "A 16384-Bit Dynamic RAM", IEEE J.S.S.C. October 1976, pp. 570-573.

22. Roger Bradley, Control Data Corporation, Minneapolis, Private Communication.

23. W. S. Richardson, "Diagnostic Testing of MOS Random Access Memories", Solid State Technology, March 1975, pp. 31-34.

24. A. Tuszynski, "Large Integrated Circuit", Coursenotes for EE 8051/2/3.

25. Mead and Conway, "Introduction to VLSI Systems", Addison-Wesley, 1980.

26. Eckhardt Wolfgang, "Electron Beam Testing of VLSI Circuits", IEEE J.S.S.C., April 1979, pp. 471-480.

27. Hiromu Fujioka, "Function Testing of Bipolar IC's and LSI's with the Stroboscopic Scanning Electrinic Microscope", IEEE J.S.S.C., April 1980, pp. 177-183.

28. DeMan et al., "A Low Input Capacitance Voltage Follower in a Compatible Silicon Gate MOS-Bipolar Technology", IEEE J.S.S.C., June 1977, pp. 217-223.

29. Apled and Gray, "A Fast-Settling Monolithic Operational Amplifier Using Doubler Compression Techniques", IEEE J.S.S.C., December 1974, pp. 332-339.

30. Burbank and Tuszynski, "Pseudo Analog Testing", 1980 IEEE Test Conference, pp. 126-130.

31. Ishikawe and Hamaguchi, "New Diagnostic Testing Technology for LSI Memories", 1980 IEEE Test Conference, pp. 225-229.

32. J. Crafts, "Bump, Bounce, and Rebound Testing", Teradyne Inc., Report #104.

33. Rao, White and Gossen, "Epitaxial layer blocks unwanted charge in MOS RAMs", Electronics, 30 June 1981, pp. 103-105.

# AUTOTESTCON '78

## INTERNATIONAL AUTOMATIC TESTING CONFERENCE

AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78
AUTOTESTCON '78

# CONFERENCE RECORD

SPONSORED BY

THE INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS
IEEE SAN DIEGO SECTION
IEEE AEROSPACE & ELECTRONIC SYSTEMS SOCIETY
IEEE INSTRUMENT & MEASUREMENT SOCIETY

PARTICIPATING SOCIETY

AMERICAN INSTITUTE OF
AERONAUTICS AND ASTRONAUTICS

# SELF-LEARNING MACHINES APPLIED TO THE TESTING OF SEMICONDUCTOR MEMORIES

A. Tuszynski
University of Minnesota at Minneapolis

## Abstract

The concept of self-learning testers is explored with reference to the detection of certain classes of soft failures. An appropriately organized machine acquires testing "know-how", while exercising memories with a virtually endless sequence of pseudo random test vectors. Verification and sorting of failures is straightforward, but imaginative compression of the acquired information is mandatory. The efficacy of the tester is, indeed, predicated on the notion of the "time trace" of a sequence of test vectors.

## Introduction

The testing of large monolithic circuits is a multifarious affair. Investigating hard failures, one has to consider the recently identified "stuck at tristate" problem (1), just as much as the old "stuck at one" and "stuck at zero" problems (2,3). But this gives just a foretaste of complications encountered when probing for soft failures. With soft failures in the offing, LSI devices cannot be tested exhaustively; willy-nilly, they must be tested intelligently (8,9). Indeed, every test must serve a distinct worthwhile purpose if a modicum of efficiency is to be achieved. Nonetheless, there is room for both deterministic and Monte Carlo techniques. The former are for anticipated weaknesses, whereas the latter are brought into play when one attempts to detect unforeseen flaws. Random errors, attributable to such perturbations as extraneous noise or nuclear radiation (10,11), are best monitored by a combination of deterministic and statistical techniques.

Among the conspicuous causes of soft failures are design deficiencies (4,5), processing flaws (6,7), and specifications malapropisms (8). This is why the synthesis of test programs for LSI devices should be preceded by in-depth studies of circuitry, topology and processing (9). Admittedly, however, while it takes time to analyze a new RAM, experimental work should commence at the earliest opportunity. Hence comes the first argument for self-learning testers as machines which can be put to work without procrastination. A second, entirely different, inducement arises from the autonomous nature of self-learning testers: being independent of human intelligence, self-learning machines are likely to reveal that which man has overlooked.

The term "self-learning" need not conjure a specter of automatons or androides (12,13); it is not proposed to bestow "artificial intelligence" upon the tester. On the contrary, the supposedly ubiquitous need for intelligence is being denied; intelligence is not considered to constitute a sine qua non prerequisite of either learning or decision making. The argument is advanced that a system, equipped with an ample memory and a sensible sorting mechanism, can acquire "know-how" which will enable it to arrive at sound decisions most of the time; it is contended that such a system will keep track of repetitive phenomena and will separate the reproducible from the random, quite efficiently.

## The Hardware

The functional components of the tester are shown in figure 1. There is a control and processing unit, a multi-phase clock, a random-number generator, an up-down counter, and there are three distinct memories, in addition to the DUT. The main memory has R sectors to accommodate as many fatal sequences, that is, sequences which disqualify at least one DUT. Each sector has two blocks, a regular RAM for initial-state data and a FIFO memory for test vectors. The reference memory is supposed to be fail-proof, by virtue of both majority voting and error correction redundance. The results memory remembers the failure record of individual DUTS. The up-down counter supplies sequential addresses on one hand, and a backtrack capability on the other. The purpose and construction of the other modules is self explanatory, but the architecture of the random number generator does deserve some elaboration.

The random number generator is built around a 45-bit shift register. Modulo two linear feedback is employed to produce "maximum length" sequences of pseudo random numbers (15,16). Running at a rate of a million words per second, the generator will furnish a one year's supply of distinct 45-bit numbers. In addition to the feedback taps, there are m+1 output terminals, one for the random "data" and m for the random address under which the data gets stored. The number of address bits is invariably smaller than 45. For this reason, some addresses will reappear and, better still, will reappear in different order at different times. This fulfills a randomness condition that has been overlooked, apparently, by the cited references. Yet another increment of entropy comes from the truly random initial state produced by means of the hardware shown in

CH1416-7/78/0000-0246 $00.75

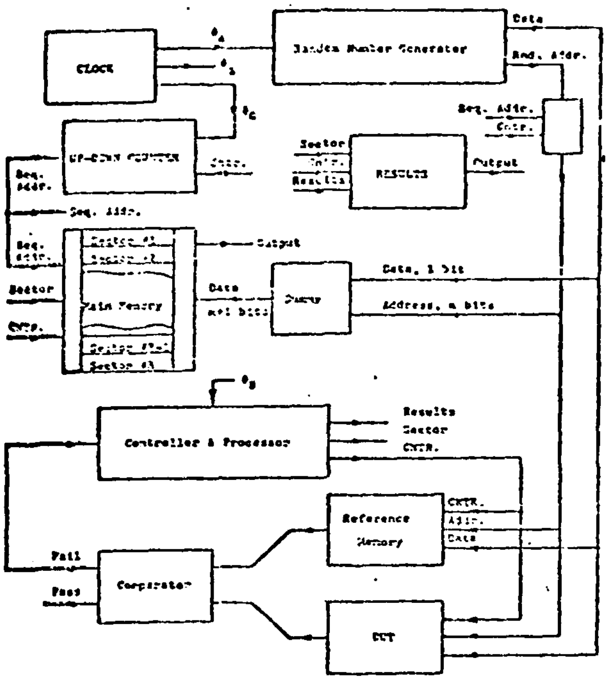


Figure 1. Components of a Self-Learning Tester

the upper part of figure 2. All in all we have in figure 2 a generator which produces an abundance of credibly random sequences.

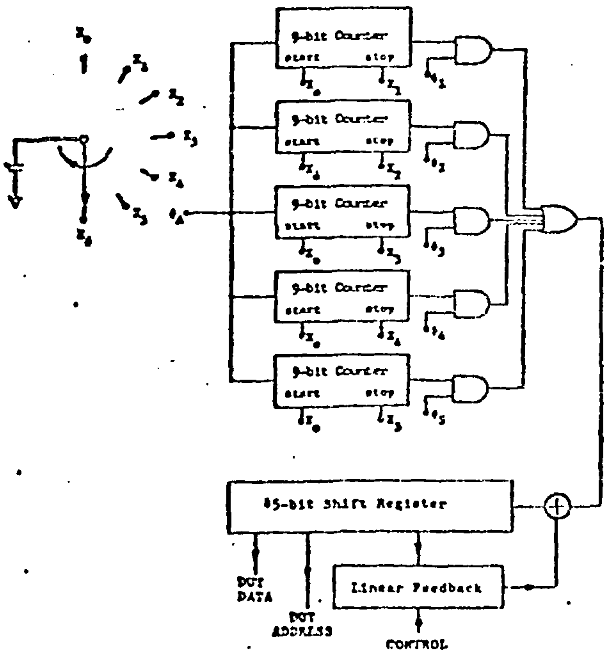


Figure 2. Random Number Generator

### Symbols and Numbers

Our DUT is a "one by M" RAM, with M taken as 65,536 whenever specific numbers are called for. There is, accordingly, one data-in pin, one data-out pin, and there are

$$m = \log_2 M = 14 \quad . \tag{1}$$

address pins. The same applies to the Reference Memory, as it must, if the DUT and the Reference device are to be operated in parallel.

The Main Memory is fairly large and complex. There are R sectors and there are two Blocks in each sector. Each Pattern Block has a capacity of M bits, just as the DUT. Each Vector Block is a FIFO memory which can store n words of m+1 bits. The number R gives the siz- of a test batch: R devices can be tested in one uninterrupted run of the machine. The number n is the yet to be discussed "trace-length" of a failure. The capacity of the Main Memory is

$$CMM = R \times M + R \times n \times (m-1) \tag{2}$$

The Results Memory keeps track of the statistics of the self-learning exercise; it remembers, for example, that DUT #3 failed at least one test in sequence #7, one in sequence #21, and one in sequence #39. Its capacity is

$$CRM = 2 \times R \times R \text{ bits} \tag{3}$$

There are two bits for each of the R tests performed on each of the R devices.

### The Range of Soft Failures

The issue of soft errors arises only where the integrity of a machine happens to be data dependent. For example, an adder is said to exhibit a soft failure, if it consistently fouls up the addition of 3.87 and 6.45, though it works correctly with all other numbers. As one would expect, a machine may manifest more than one soft error, and such errors may be either related or independent. Soft errors do come in all shapes and sizes, being brought about by various, more or less obscure causes; classification of soft failures is frequently difficult, and attempts at generalization often lead to confusion. Our immediate concern is, fortunately, limited to the distinction between Pattern Sensitivity and Sequence Sensitivity as it applies to memories, and memories only.

Since an ordered set of M bits has a range of

$$W = 2^M \tag{4}$$

numbers, the storage matrix of a 64-kbit memory can be arranged into

$$W(64k) = 2^{65,536} \approx 10^{19,728} \tag{5}$$

distinct patterns of ones and zeros. Working at a rate of a million tests per second, one performs

$$Y(365) = 3.15*10^{13} \tag{6}$$

tests per year. To check all patterns which enter into equation 4, one would work

$$Y(W) = 3*10^{19,724} \text{ years} \qquad (7)$$

These are illusory numbers that fail to enlighten the querist. Yet, they show up quite frequently in LSI engineering, since their appearance is by no means limited to large memories. To check all patterns of a modest 256-bit RAM at a rate of a million tests per second, one would need

$$Y(256) = 3*10^{64} \text{ years} \qquad (8)$$

and, as already noted, 45 bits still take a full year. The concept of exhaustive testing has no place in LSI devices. Exhaustive testing ends at about 30 bits, since it takes 33 minutes to check out a billion patterns at a rate of 1 MHz.

Pattern Sensitivity versus Sequence Sensitivity

The expression "Pattern Sensitivity" refers to failures associated with individual patterns of data. "Sequence Sensitivity", on the other hand, refers to failures brought about either by execution of specific sequences of operations or by processing of specific sequences of data patterns. Speaking of one particular failure, a single data word is called out for pattern sensitivity, but an entire ordered set of either instructions or data words is listed in the case of sequence sensitivity. For example, a pattern sensitivity failure of a four bit memory might be reported as

$$f_i(9) = ps_i(1001) \qquad (9)$$

but Sequence Sensitivity must necessarily show up as a string of instructions or data words:

$$f_j(9) = ss_j(1001/0001/0011/0111/\ldots) \qquad (10)$$

Referring to sequences of "states" rather than sequences of "operations" or "data words", it is convenient to write a sequence sensitivity statement as:

$$f_k = ss(X_{k,o}/X_{k,-1}/X_{k,-2}/\ldots/X_{k,-n}) \qquad (11)$$

where $X_{k,o}$ is state zero, that is, the state which reveals the existence of a problem, while $X_{k,-1}$ is the first preceding state and $X_{k,-n}$ is the n-th preceding state. Equation 11 implies that, having observed a failure, we backtrack n steps towards the source of trouble. The problem is supposed to be cumulative rather than discrete, $X_{k,-n}$ being the furthermost significant state. A classical sequence sensitivity failure will recur, if the given sequence is replayed, beginning at state $X_{k,-n}$. Equation 11 represents the "trace" of sequence number "k"; subscript "n" gives the length of the trace in terms of machine cycles, while

$$L = nT \qquad (12)$$

where T = period of a machine cycle

gives the length of the trace in seconds. It remains to estimate n and L but, before we do this, it may be advisable to approach the distinction between pattern and sequence sensitivity from an alternate direction.

The phrase "a failure is associated with state Xo" is somewhat incomplete, since a failure may be associated with attempts to either enter, or leave, or maintain a given state. Only the last of these possibilities can be described adequately by a one-word quotation of the offending pattern, and only this possibility constitutes a static or, more precisely, a quasi-static situation. Having written some data into the memory, we simply want to keep it there and we generally succeed, except for the peculiar case of one specific data word which tends to degenerate because of leakage or "refresh" noise. This is pattern sensitivity. However, in contrast to the just discussed static case, the other two failures are dynamic in nature. We may be unable to either change $X_{i,-1}$ into $X_{i,o}$ or $X_{i,o}$ into $X_{i,+1}$. For example, a one-by-four RAM may manifest a failure when an attempt is made to go

$$\text{from} \quad X_{i,-1} = 1000 \qquad (13a)$$

$$\text{to} \quad X_{i,o} = 1001 \qquad (13b)$$

though the transition

$$\text{from} \quad X_{i,-1} = 1011 \qquad (13c)$$

$$\text{to} \quad X_{i,o} = 1001 \qquad (13d)$$

is invariably sound. Here we deal with a case of sequence sensitivity that is particularly convincing if the said failure is predicated on the specific values

$$X_{i,-2} = 0001 \qquad (14e)$$

$$X_{i,-3} = 0000 \qquad (14f)$$

$$\ldots\ldots\ldots\ldots$$

$$\text{and} \quad X_{i,-n} = 1010 \qquad (14n)$$

recorded as

$$ss_i(9) = 1001/1011/0001/0000/\ldots/1010 \qquad (15a)$$

$$\text{or} \quad ss_i(9) = 9/11/1/0/\ldots/10 \qquad (15b)$$

The distinction between pattern sensitivity and sequence sensitivity may be minimal. It is theoretically possible, though practically unlikely, that attempts to write $X_{i,o} = 9$ will fail irrespectively of the choice of $X_{i,-1}$, $X_{i,-2}$, etc. One can classify such cases as pattern sensitivity because a single data word, i.e.,

$$f_i = ps_i(9,\text{write}) \qquad (16)$$

does describe the failure adequately. However, one can also classify them as degenerate sequence sensitivity because transitions are at issue, though n is equal to one in the statement

$$f_1 = ss_1(-/9) \qquad (17)$$

Regular sequence sensitivity is, of course, characterized by

$$n > 1 \qquad (18a)$$

or even $n >> 1$ (18b)

The number n may vary widely. We do not concern ourselves with its exact value, but we must estimate the upper bound of this important number.

### The Length of the Trace of a Failure

The present state of a discrete-time system can be written in conventional notation (9) as

$$X(0) = BU(-T) + CX(-T)e^{-A(T/\tau)} \qquad (19a)$$

$$= BU(-T) + BCU(-2T)e^{-A(T/\tau)} + C^2B(-3T)e^{-A(2T/\tau)} +$$

$$+ C^nX(-nT)e^{-A(nT/\tau)} \qquad (19b)$$

X(0) evidently depends first on the most recent stimulus, then on the situation at $t = -T$, and only eventually on events at $t = -nT$. Recollection of past events fades away quite rapidly, as long as the circuitry is in a conducting state. The relative importance of any particular state is attenuated by 80 dbs in a time interval

$$L_A = nT/\tau = 10*(\text{Access Time}) \qquad (20)$$

A complication arises, however, because storage capacitors of "single-transistor" RAMS are maintained in a non-conducting state, most of the time. Time constants of cut-off circuitry may be as long as 30 minutes or even more (17). Nonetheless, this is only superficially disconcerting. The cut-off state of dynamic storage elements is periodically interrupted by the "refresh" mechanism. We are free, therefore, to think of a sequence of refresh operations and to assert that the effect of any single event wears off in time $L_R$ equivalent to 10 refresh periods:

$$L_R = 10*T_R \qquad (21)$$

Most manufacturers of semiconductor memories stipulate a refresh time of 2 milliseconds, and most users abide by this requirement. We can assume, consequently, that the trace of a failure does not exceed 20 milliseconds, even where storage capacitors enter into the act.

### Procedures

Each DUT is initialized with its own random data; addresses are called out sequentially by the Up-Down Counter, but the "data bit" is supplied by the Random Number Generator. The binary states of the data bit are interpreted as "change" and "no change" rather than as "one" and "zero". All initialization data is fed simultaneously to the DUT, the Reference Memory, and the appropriate sector of the Main Memory. A read operation follows in the wake of each write operation beginning with address 1 and ending with address M. Detection of a failure diverts the tester into a more complex initialization procedure, without prejudice to the already acquired failure information. One way or another, by the time address M is reached, the state of the DUT is duplicated in the Main Memory as well as in the Reference Memory; this state of the DUT is denoted by $X_I$ for linguistic convenience.

In the case of the first DUT, random testing begins immediately after initialization. Test vectors furnished by the Random Number Generator are fed simultaneously to the DUT, the Reference Memory and sector 1 of the Main Memory. The Pattern Block of sector 1 follows the DUT, while the Vector Sector stores the individual test vectors in FIFO fashion. When a failure is detected, the information in the Vector Sector is used to restore the system to either state $X_I$ or $X_{-n}$, whichever may apply. The trace of the failure is then rerun through the DUT and the Reference Memory for verification purposes. We need to know whether the observed failure can be reproduced at will. The Pattern Sector does not participate in the second phase of the verification exercise; it just retains the "initial pattern" information -- either $X_I$ or $X_{-n}$ -- for use on other DUTS.

When DUT #2 is tested, it is initialized, tested for susceptibility to the trace which failed DUT #1, and only then does random testing begin. It continues till DUT #2 fails and the results of the failure are duly recorded in sector number two of the Main Memory and location number two of the Results Memory. DUT #3 is initialized, checked out for susceptibility to traces # 1&2, and then exercised with random vectors. DUT #W is tested for susceptibility to W-1 previously generated traces before it gets a chance to generate its own trace. Results of all tests are recorded in the Results Memory.

### Cross-References

Cross-references help with the diagnostics of a given family of devices. As far as traces generated by other devices are concerned, a DUT may pass a test, fail during loading, fail on completion of all n steps of a trace, or it may fail early. It may also be either able or unable to reproduce the failure spelled out by its own trace. We want to know how many devices displayed unconfirmed failures, how many failed early, and so on.

Purely random failures are not reproducible. Nuclear radiation problems, if any exist, should generate a fair number of "unconfirmed" errors. Needless to say, however, there exist many causes of random errors. Convincing evidence of randomness is a necessary, though highly insufficient, prerequisite of arguments which seek to attribute device failures to background radiation.

Fabrication-process flaws come in many guises. Pinhole-size flaws vary in position from chip to chip and from wafer to wafer. Patch-size flaws, on the other hand, tend to show up in the same section of the wafer; a significant fraction of all chips may display the same type of flaw and the same type of failure. In any case, be they pinhole or patch size, processing flaws are generally reproducible; they do not generate "unconfirmed" failures.

Circuit design flaws are highly reproducible. They are certainly permanent, and they are common to all chips, though they may show up only in response to one particular sequence. Yield is likely to be low. If one particular trace disqualifies many chips, the possibility of a circuit design flaw should be investigated, before time is spent on other explanations.

All in all then, random phenomena bring about unconfirmed faults, and processing flaws generate fixed failures in a finite fraction of all chips, while circuit design delinquencies generally result in low yields.

## Conclusion

A self-learning tester is feasible: A CACHE arrangement with a disc and a fair-sized semiconductor memory will service both the Main Memory and the Result Memory. All other hardware requirements are quite modest.

Apart from generating useful sequences of test vectors, a self-learning tester will accumulate statistics for diagnostic endeavors. It will separate random perturbations from fixed flaws, and will supply information for yield analysis. With luck, it may even uncover complications overlooked by the designer.

It has, of course, its limitations. Evaluation of LSI devices calls for thorough analysis, exploratory experimentation, hard-failure testing, and random-testing. Self-learning testers are only for random testing and certain aspects of exploratory testing.

## References:

1. R. L. Wadsack, "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits", BSTJ, May-June 1978, pp. 1449-1488

2. Chang, Manning and Metze, "Fault Diagnosis of Digital Systems", John Wiley, 1970

3. Friedman and Menon, "Fault Detection in Digital Circuits", Prentice Hall, 1971

4. W. S. Richardson, "Diagnostic Testing of MOS Random Access Memories", Solid State Technology, March 1975, pp. 31-34

5. J. Reese Brown, Jr., "Timing Peculiarities of Multiplexed RAMS", Computer Design, July 1977, pp. 85-92

6. Sie, Youngblood, Liao and Turk, "Soft Failure Modes in MOS RAMS", IEEE Reliability Physics Symposium, 1977, pp. 27-30

7. T. C. Lo and Mark Guidry, "An Integrated Test Concept for Switched Capacitor Dynamic MOS RAMS", IEEE JSSC, Dec. 1977, pp. 693-703

8. George P. Nelson, "Microprocessor Testing and Performance Quantification", International Symposium on Military & Industrial Microprocessor Systems, June 1975

9. D. D. Rinerson & A. Tuszynski, "Identification of Causes of Pattern Sensitivity", IEEE Test Symposium 1977, pp. 166-170

10. T. C. May and M. H. Woods, "A New Physical Mechanism for Soft Errors in Dynamic Memories", Intel Corporation, 1978

11. J. C. Pickel & J. T. Blandford, "Cosmic Ray Induced Errors in MOS Memory Cells", IEEE Conf. on Space Radiation Effects, 1978

12. Robert G. Middleton, "Computers and Artificial Intelligence", Howard W. Sams & Co., 1969

13. John Formby, "An Introduction to the Mathematical Formulation of Self-Organizing Systems", Van Nostrand, 1965

14. Fogel, Owens and Walsh, "Artificial Intelligence Through Simulated Evolution", John Wiley, 1966

15. S. W. Golomb, "Shift Register Sequences", Holden Day, 1969

16. R. A. Frohwerk, "Signature Analysis", Hewlett-Packard Journal, May 1977, pp. 2-8

17. F. J. Morris and T. A. Shankoff, "Charge Coupled Device Processing", Electrochemical Society Extended Abstracts, 1976, Abstract No. 196

# Medium and Soft Errors

Al Tuszynski

University of Minnesota

IEEE Workshop on Memory Testing, 10 November 1980

Classification of Errors

The JC-42-78-65 publication of the Join Electron Device Engineering Council (JEDEC) classifies errors into three categories, namely hard, medium and soft. Hard Errors are manifested by cells which "will not properly store data under any condition of test operation". Medium Errors are said to exist when "cells will properly store data under some conditions of operation (voltage, temperature, timing and pattern), but will make errors under other conditions of operation". Soft Errors correspond to "an occasional random loss of data. The reason for the error is normally not identifiable, nor is it repeatable on a device tester".

Our own preference favors a straight binary division into hard and soft errors only. "Hard" errors are attributable to "hardware" defects such as shorts, missing contacts and floating gates. "STUCK-AT" faults are classical examples of hard errors; if a node is "STUCK-AT-ZERO" it will be low regardless of input data. In that sense one can say that the manifestation of hard errors is data independent. Soft errors on the other hand appear only in the wake of certain patterns and sequences of data or addresses. They are attributable to design and processing flaws as much as to environmental perturbations.

In brief, we distinguish merely between data-independent and data-dependent errors. Limited in scope as this classification may be, we need it to articulate a comprehensive strategy of testing. The word data, as used here, refers both to addresses and to input information.

## Classification of Soft Errors

No classification is more important than the division into random and non-random errors. The latter can be reproduced on command but the former cannot! This is indeed how we can and should tell them apart. When an error is observed one ought to jump back sufficiently far in the test program to guarantee re-emergence of reproducible errors. Observations of failures must be verified and diagnosed if the testing of VLSI devices is to become effective. Test programs should include appropriate retrace loops.

Nuclear radiation perturbations are among the most publicized causes of random errors. Faced with them, one ought to determine their rate of occurrence. Even more important is the question of single or burst errors. Single errors can be corrected quite easily, but bursts cannot. Cosmic rays are potentially more dangerous than alpha particle. Polyimide coating of the chip is used nowadays to guard against radioactive materials imbedded in the package. The range of alpha particles is limited to 100 microns. Cosmic rays, on the other hand, cannot be stopped, not unless one is willing to use ten centimeters of lead.

Reproducible soft errors are mostly a matter of design or fabrication flaws. The circuitry of modern memories is complex; there may be performance marginalities in the address decoders, or the output section or, as is frequently the case, the bit-line circuitry. Figure 1 illustrates the interdependence of the many elements associated with the sense amplifiers of dynamic RAMs.

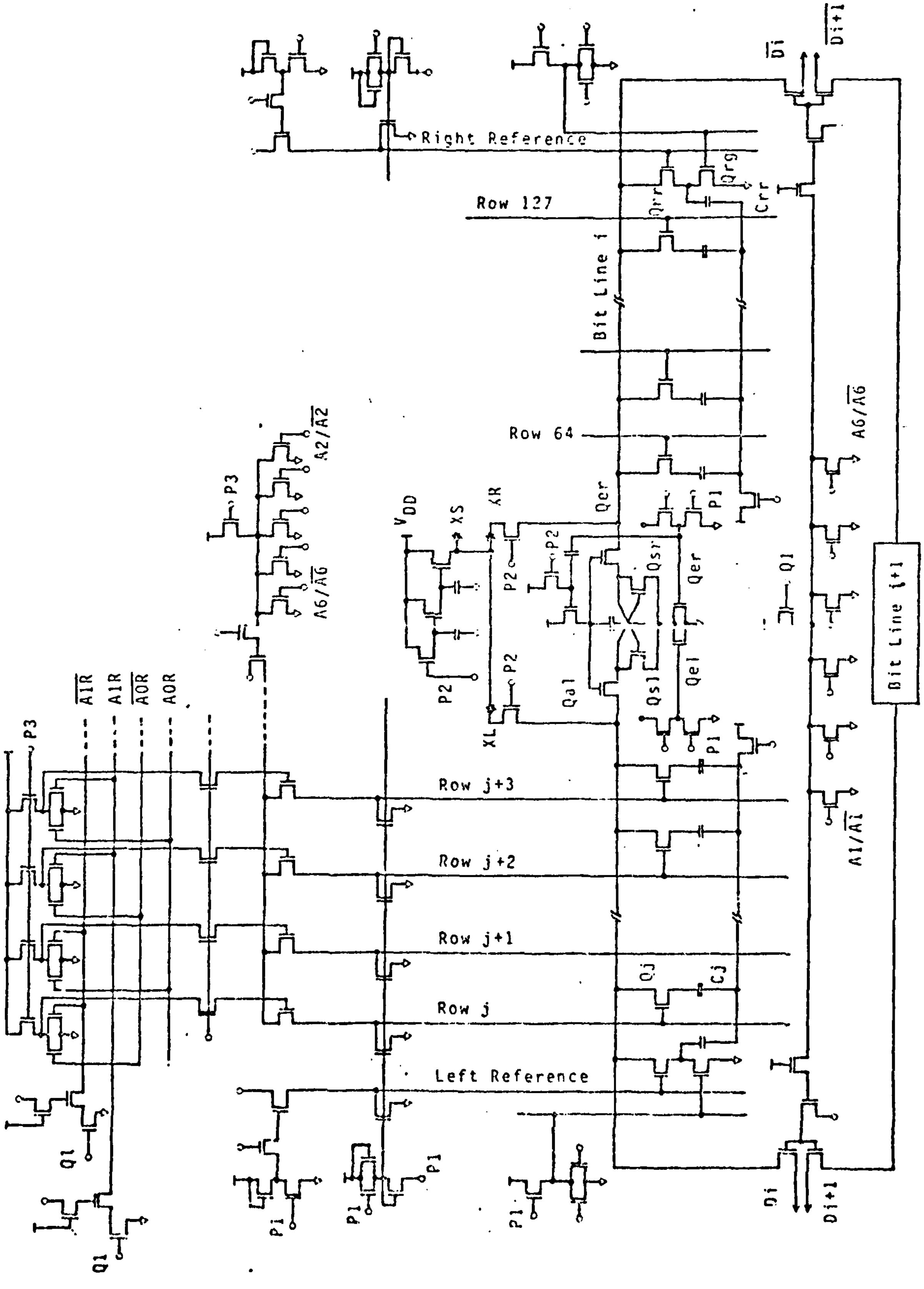


Fig.1: Bit-Line Circuitry

## Commercial Test Patterns

Various test programs were developed by vendors and users of semiconductor memories. Manufacturers of test equipment contributed their share too. More or less accurate claims regarding the effectiveness of individual programs are based on empyrical data rather than mathematical arguments. They should be treated with caution. All reasonably good programs take care of hard errors; the justification of extra cost comes from soft errors. These differ, however, from one product to the next and one vendor to another. Consequently, there are no universally good programs. To be cost effective, a program must be personalized with respect to a particular product and vendor with attention to modifications continually introduced by memory manufacturers in the interest of yield.

It takes, of course, time to analyze a new product and to develop an appropriate program. One cannot stop the world while waiting for the ideal test to emerge. That is where commercial programs play a useful role in spite of their limitations. We give here three examples, not necessarily the best. The total number of operations may be proportional to $N^1$ or $N^{3/2}$ or $N^2$. Our purpose is to demonstrate the differences between these three categories of tests. Our programs are written in self-explanatory pseudo-fortran language.

We begin with the well known MARCH program.

| March | # of operations |
|---|---|
| Reset the background to ZERO | N |
| For i=0 to N-1 | |
| Read ZERO | |
| Write ONE | |
| Return | 2N |
| For j = N-1 to 0 | |
| Read ONE | |
| Write ZERO | |
| Return | 2N |
| Reset the background to ONE | N |
| For i=0 to N-1 | |
| Read ONE | |
| Write ZERO | |
| Return | 2N |
| For j = N-1 to 0 | |
| Read ZERO | |
| Write ONE | |
| Return | 2N |
| Total number of cycles | 10N |

```
Galpat                                                    # of operations

A.  Reset background to ZERO                                     N

    Write ONE into cell 0
    For i=1 to N-1
    Read cell 0
    Read cell i
    Return
    Write ZERO into cell 0                                      2N

    For i=1 to N-2
    Write ONE into cell i
    For j=0 to i-1 and j=i+1 to N-1
    Read cell i
    Read cell j
    Return
    Write ZERO into cell i
    Return                                                      2N(N-2)

    Write ONE into cell N-1
    For i=0 to N-2
    Read cell N-1
    Read cell i
    Return
    Write ZERO into cell N-1                                    2N

    Half of the total number of cycles                          $2N^2+N$

B.  Reset background to ONE

    Run program similar to part A with appropriate change from
    ZERO to ONE and vice versa

    Total number of cycles                                      $2(2N^2+N)$
```

## GALTDIA

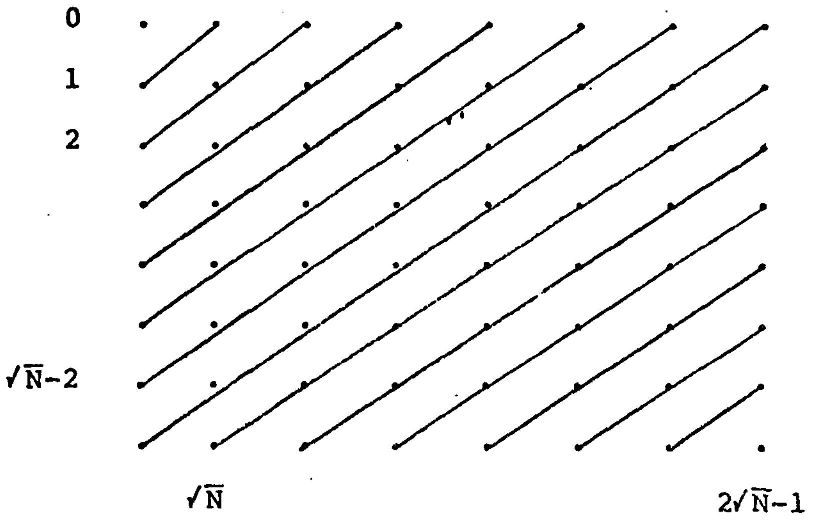


Fig. 1: Pertinent Diagonals

To compute the total number of operations performed in this program we will need the formulas for the sum of natural numbers. The required expressions can be written as

$$S1 = 1+2+3+\ldots r \tag{1a}$$

$$= \sum_{i=1}^{r} i \tag{1b}$$

$$= (1/2)r(r+1)$$

and

$$S2 = 1^2+2^2+3^2+\ldots r^2 \tag{2a}$$

$$= \sum_{i=1}^{r} (i)^2 \tag{2b}$$

$$= (1/3)[(r+1)^3 - (1/2)(r+1)(3r+2)] \tag{2c}$$

$$= (1/3)r(r - \frac{1}{2})(r-1) \tag{2d}$$

As to the details of the GALTDIA program, refer to figure 1 and proceed as follows:

```
A.  Reset the field to ZERO

20  For $i=1$ to $2\sqrt{N}-2$
    Go to diagonal $i$
    For $j=0$ to $i$
    Select a test cell
    Read the previous cell
    Read the test cell
    Complement the test cell
    Gallop along the diagonal
    Complement the test cell
    Return
120 Return

B.  Rcset the field to ONE

    Repeat lines 20 through 120
```

To proceed with the computational aspects of GALTDIA, let us go through the following observations:

$$(\text{No. of cells in diagonal } i) = i+1 \tag{3}$$

$$\text{No. of gallop operations per test cell} = 2i \tag{4a}$$

$$\text{No. of conditioning opertions per test cell} = 4 \tag{4b}$$

$$\text{Total No. of operations per test cell} - 2(i+2) \tag{4c}$$

$$\text{No. of operations per diagonal} = 2(i+1(i+2) \tag{5a}$$

$$= 2i^2+6i+4 \tag{5b}$$

$$\text{No. of operations in a Reset step} = N \tag{6}$$

Now let Q be the total number of cycles in a GALTDIA program. Noting the numerical duplication between diagonals 1 through $\sqrt{N}-2$ on one hand and $\sqrt{N}$ through $\sqrt{N}-2$ on the other, one can write Q as

$$Q = 2\{\sum_{i=1}^{\sqrt{N}-2} 2[2(i)^2+6i] + 8(\sqrt{N}-2) + 2\sqrt{N}(\sqrt{N}+1)+N\} \tag{7a}$$

Substitution of equations 1 and 2 into 7a leads immediately to

$$Q = 2\{\tfrac{4}{3}(\sqrt{N}-2)(\sqrt{N}-\tfrac{3}{2})(\sqrt{N}-1)+6(\sqrt{N}-2)(\sqrt{N}-1)+3N+10\sqrt{N}-16\} \tag{7b}$$

$$= 2(\tfrac{4}{3}\,N^{3/2} + 3N + \tfrac{2}{3}\,N^{1/2} - 8) \tag{7c}$$

## Testing of VLSI Hardware

Small circuits can be tested exhaustively; large circuits cannot. Soft errors are responsible for the dilemma. Being data dependent, they may show-up ih the wake of but a few combinations of input and address bits. Manifestations described vividly as pattern and sequency sensitivity are indeed quite common. Many problems arise also from inadequate identification of worst-case conditions. As it happens, memories pose more problems than microprocessors.

To demonstrate the futility of massive testing consider simple pattern sensitivity in a small memory and let your intuition translate the results into large-memory language.

N bits can be arranged into

$$C(N) = 2^N \tag{1}$$

different combinations such as

$$\begin{array}{ccccccccc} 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 & 1 \\ 0 & 1 & 0 & . & . & . & 1 & 0 & 1 \\ - & - & - & - & - & - & - & - & - \\ 1 & 1 & 1 & . & . & . & 1 & 1 & 1 \end{array} \tag{2}$$

to examine each, one would have to perform at least

$$T(N) = 2^{(N+1)} \tag{3}$$

tests. This would add up to

$$T(1024) = 2^{1025} \tag{4}$$

$$\approx 10^{300} \tag{5}$$

tests, in the case of a 1k memory. Working at the rate of ten million test per second, one would need

$$S(1024) \approx 10^{293} \text{ seconds} \tag{6}$$

$$\approx 3 \times 10^{285} \text{ years}$$

to complete the test.

We need not go any further. An intelligent alternative to exhaustive testing is obviously called for.

## Test Strategy

Chip design, system architecture, testing and reliability are inter-dependent. The testability objective imposes - when it comes to memories - two conditions on system architecture.

1. In a m-by-N memory, use 1-by-N chips in preference to m-by-(N/m) chips (see figure 2); the former are less likely to produce multiple errors.
2. Use one bit error correction. This is not half as expensive as it appears to be. The correction code can be applied to multiple words in order to reduce the overhead burden (see reference 13).

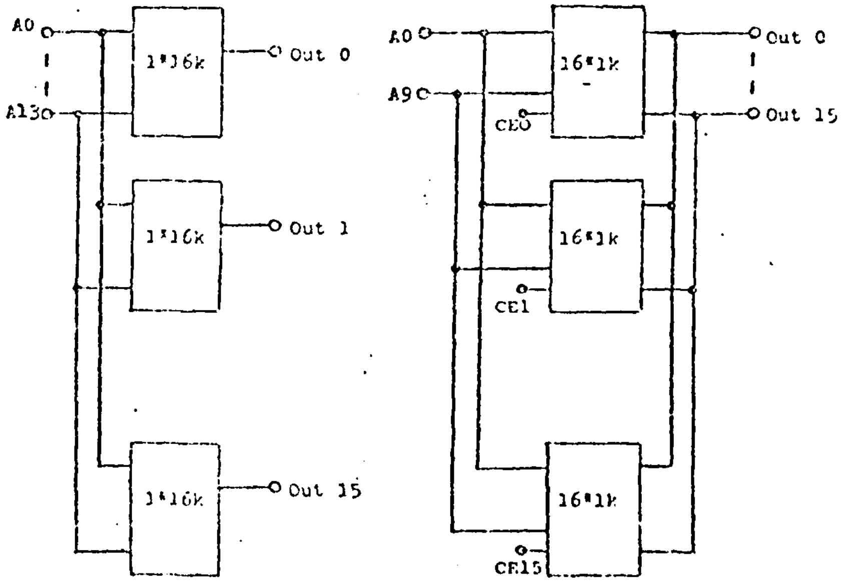
The actual test is organized into three phases, the last two of which can be combined into one.

1. parametric testing
2. hard-error testing by means of general purpose programs
3. soft-error testing by means of personalized programs

Parametric testing, that is the measurement of supply currents, input capacitances etc. identifies bad units very efficiently. It can also be used to monitor the consistency of the fabrication process and thus issue warnings of soft-error hazards.

The hard-error test is supposed to ensure that 1) each cell will properly accept, store and return a HIGH or a LOW and 2) the decoding circuitry actually yields N distinct addresses. The primitive program of figure 3 will accomplish both these tasks. [illegible] programs will do that also but by more ambitious

Programs for the detection of soft errors should come from intensive theoretical and experimental study. Every product and every modification of every vendor must be investigated. Mature products display fewer - and possibly different - errors than their predecessors. Once the causes of marginal performance are known, suitable programs can be written and combined with the hard-error complements. Commercial (and therefore general purpose) programs attempt to do just that, but their cost effectiveness is poor indeed. Only $N^1$ tests make sense when it comes to large chips (16K and up); $N^2$ and even $N^{3/2}$ programs are completely inacceptable. There is no choice: One must analyze the design and probe the chip till all significant sources of soft errors are identified and eliminated. To be feasible, test programs for soft-errors must be personalized. Furthermore it is necessary to base test strategies on the premise that in devices which have passed the hard-error test, soft-errors are only few and far between.



a) Good

b) Bad

Fig.2: Architecture of a 16-by-16k Memory

## The Cost of Error Correction

At least two items must be identified when error correction is contemplated: 1) The access-time penalty and 2) the cost of excess hardware. Dynamic response cannot be quantized without reference to both system details and chip characteristics; a lot depends on the difference between access-time and cycle-time - pipelining can be used to shift the focus from the former to the latter. The excess hardware, on the other hand, can be projected in general terms.

To assess the cost or error-correcting-hardware, let there be m data bits and k correction bits for a total of m+k. There may be an error in any one of the m+k bits. Required is, therefore, a message which states:

there is no error<br>
or there is an error in bit one<br>
or there is an error in bit two

---

or there is an error in bit m+k

That adds-up to

$$M = m+k+1 \quad (1)$$

messages which are to be articulated by means of k bits.

There is a repertiore of

$$K = 2^k \quad (2)$$

messages in k bits. What one wants is

$$K > M \qquad (3)$$

and, therefore

$$2^k > m+k+1 \qquad (4)$$

i.e.

$$2^k-k > m+1 \qquad (5)$$

Table I confirms what was to be expected: the cost of excess hardware is acceptable if one operates on sufficiently long words. Consequently, one might consider error correction for multiple rather than single words and that is exactly what is done in IBM system 370/158 (reference 13 of Test Strategies). Application to memories of block-oriented error correction is a viable research topic.

Nota bene that we talk about off-chip error correction. On-chip correction should not be attempted without prior attention to many outstanding physical and architectural details.

Table 1: Single Bit Error Correction

| m | k | % efficiency |
|---|---|---|
| 1 | 2 | 33 |
| 2 | 3 | 40 |
| 4 | 3 | 57 |
| 8 | 4 | 67 |
| 16 | 5 | 76 |
| 32 | 6 | 84 |
| 64 | 7 | 90 |
| 128 | 8 | 94 |

## Conclusion

Less is better, when it comes to testing. Invest in research rather than test equipment. Massive testing is expensive yet unproductive. Emphasis must be put on the elimination of multiple errors rather than the detection of every individual error. That is why perusal of circuit design is crucially important. Design flaws lead to the inacceptable multiple errors. Testability - in the case of memories - means 1) architecture per figure 2 and 2) single-error correction.

Hard failures can be caught easily. Soft errors should be rare; single-error correction will eliminate them. A continual diagnostic effort must be maintained where reliability is of a premium.

Parametric testing can provide clues and warnings. The consistency of the manufacturing process must be monitored by the user as well as the manufacturer. In VLSI testing, there is no substitute for know-how.

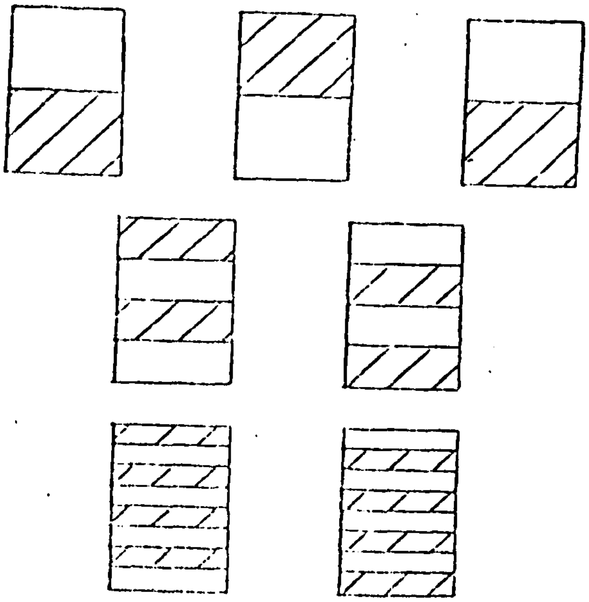


Fig.3: Test Patterns for an 8-bit RAM

# References

## Test Strategies

1. Warren G. Fee, "Memory Testing", IEEE Tutorial, Memory Testing, Catalog No. EHO 122-2, 1977.

2. Richard Seltzer, "Test Strategies for LSI and Memory", Circuit Manufacturing, pp. 40-46, March 1977.

3. W. S. Richardson, "Diagnostic Testing of MOS Random Access Memories", Solid State Technology, pp. 31-34, March 1975.

4. John Cocking, "RAM Test Patterns and Test Strategy", IEEE Tutorial, Memory Testing, Catalog No. 122-2, 1977.

5. John P. Hayes, "Detection of Pattern-Sensitive Faults in Random-Access Memories", IEEE Trans. on Computers, pp. 150-157, February 1975.

6. R. P. Davidson, "Some Straightforward Guidelines Help Improve Board Testability", EDN, pp. 127-129, May 5, 1979.

7. James E. Fisher, "4k RAMs: Increased Densities Bring Difficult Testing Problems", EDN, pp. 47-51, November 20, 1974.

8. Robert Sugarman, "Schmoo-Hunting Becomes an Expensive Sport", Electronic Engineering Times, May 1976.

9. Derrell Coker, "16k RAM Eases Memory Design for Mainframes and Minicomputers", Electronics, pp. 115-119, April 23, 1977.

10. J. Reese Brown, Jr., "1103 Semiconductor Memory Device Pattern Sensitivities and Error Modes", 1973 LSI Test Symposium, pp. 63-76.

11. J. Reese Brown, Jr., "Timing Peculiarities of Multiplexed RAMs", Computer Design, pp. 85-92, July 1977.

12. A. Tuszynski, "Self-Learning Machines Applied to the Testing of Semiconductor Memories", IEEE Autotestcon '78, pp. 246-250.

13. Frank Bartocci, "A Microprogrammed Storage Diagnostic", University of Minnesota, M.S. Paper, 1980.

14. Lewandowski, Nelson and Orr, "An Approach to 64k RAM Testing", Electro '79, Paper #20/2.

15. Lo and Guidry, "An Integrated Test Concept for Switched-Capacitor Dynamic MOS RAMs", IEEE J.S.S.C., pp. 693-703, December 1977.

16. James M. Crafts, "Techniques for Memory Testing", Computer, pp. 23-31, October 1979.

## Radiation Effects

1. Samuel Glasstone, Sourcebook on Atomic Energy, D. Van Nostrand, 1950.

2. Mark Brodsky, "Hardening''RAMs Against Soft Errors", Electronics, 24 April 1980, pp. 117-122.

3. Kato, Koxiamoto, Mitsusada and Itoh, "64k RAM Rebuffs External Noise", Electronics, 31 July 1980, pp. 103-106.

4. May and Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories", IEEE Trans. Electron Devices, Jan. 1979, pp. 2-9.

5. W. R. Iverson (Report on views of Patrick J. Veil of RADC), "Do Cosmic Rays Spell Death for VLSI?" Electronics, 22 November 1979, pp 44-46.

6. Pickel and Blandford, "Cosmic RAY Induced Errors in MOS Memory Cells", 1978 IEEE Conference on Nuclear and Space Radiation Effects, paper #X78-3171501.

7. Raymond P. Caprice, "Alphas Stymiestetics", Electronics, 15 March 1979.

8. Editorial, "Polyimide Prevents Alpha Errors", Electronic Design, 1 September 1980, p32.

9. L.W. Rickets, "Radiation Effects on Microelectronic Components and Circuits, Solid State Technology, April 1972, pp. 50-55.

## Test Patterns

1. J. Reese Brown, Jr., "Pattern Sensitivity in MOS Memories", IEEE Test Technology Conference 1972, pp. 33-46.

2. Gleser and Subak-Sharpe, Integrated Circuit Engineering, Addison-Wesley 1977, para 11.9.

3. "Selecting Test Patterns", Macrodata Corporation, Application Note 139, April 1977.

4. De Jonge and Smulders, "Moving Inversion Test Pattern is Thorough, Yet Speedy", Computer Design, May 1976, pp. 169-173.

5. Lewandowski, Nelson and Orr, "An Approach to 64k RAM Testing", Electro 79, paper 20/2 pp. 1-5.

6. Chian and Standridge," Pattern Sensitivity in 4kRAM Devices", Computer Design February 1975, pp. 88-90.

## Catalog of Faults

1. Paul Roth, Computer Logic, Testing and Verification, Computer Science Press, 1980.
2. Chang, Manning and Metze, Fault Diagnosis of Digital Systems, John Wiley, 1970.
3. Breuer and Briedman, Diagnosis of Reliable Design of Digital Systems, Computer Science Press.
4. J. Reese Brown, "Timing Pecularities of Multiplexed RAMs" Computer Design, pp. 85-91, July 1977.
5. Lo and Guidry, "An Integrated Test Concept for Switched-Capacitor Dynamic MOS RAMs", IEEE J.S.S.C., pp. 693-703, December 1977.
6. Sie, Youngblood, Lio and Turk, Soft Failure Modes in MOS RAMs, Burroughs Corp., Plymouth, Michigan 48170, (313-453-1400).
7. Jason P. Srini, "API Tests for RAM Chips", Computer, pp. 32-36, July 1977.
8. Rinerson and Tuszynski, "Identification of Causes of Pattern Sensitivity", 1977 Semiconductor Test Symposium, pp. 166-170.

## Test Instrumentation

1. Eckharld Wolfgang et al., "Electron Beam Testing of VLSI Circuits", J.S.S.C., pp. 471-480, April 1979.
2. Hans-Peter Feuerbaum et al., "Quantitative Measurement with High Time Resolution of Internal Waveforms on MOS RAMs Using a Modified Scanning Electron Microscope", J.S.S.C., pp. 319-325, June 1978.
3. Hiromu Fujioka, "Function Testing of Bipolar IC's and LSI's with the Stroboscopic Scanning Electronic Microscope", J.S.S.C., pp. 177-183, April 1980.
4. John Gosh, "Electron Beam Harmlessly Probes High-Density Chips", Electronics, pp. 65-66, July 31, 1980.
5. John Gosh, "E Beam Makes Surface Acoustic Waves Visible", Electronics, pp. 65-66, August 28, 1980.
6. Perkin-Elmer ETEC Inc., "Autoscan V200", (415)783-9210, P. J. Breton.
7. DeMan, Vanparys and Cuppens, "A Los Input Capacitance Voltage Follower in a Compatible Silicon-Gate MOS-Bipolar Technology", IEEE J.S.S.C., pp. 217-223, June 1977.
8. Burbank and Tuszynski, "Pseudo Analog Testing", IEEE 1980 Test Conference, pp. 126-130.
9. Ishikawa and Hamaguchi. "New Diagnostic Testing Technology for LSI Memories", IEEE 1980 Test Conference, pp. 225-229.

## Error Correction

1. R.T. Chien, "Memory Error Control: Beyond Parity", IEEE Spectrum, July 1973, pp. 18-23.

2. A.V. Ferris-Prabtu, "Improving Memory Reliability Through Error Correction", Computer Design, July 1979, pp. 137-144

3. Robert Korody and David Raaum, "Purge your memory array of pesky error bits", EDN, May 1980, pp. 153-158.

4. Robet Swanson, "Matrix Technique leads to Direct Error Code Implementation", Computer Design, August 1980, pp. 101-108.

5. Richard C. Montgomery, "Simple Hardware Approach to Error Detection and Correction", Computer Design, Nov. 1978, pp. 109-118.

6. Bryan Rickard, "Automatic Error Correction in Memory Systems", Computer Design, May 1976, pp. 179-182.

7. Dusty Morris, "ECC Chip Reduces Rate in Dynamic RAMs", Computer Design, Oct. 1980, pp. 137-142.

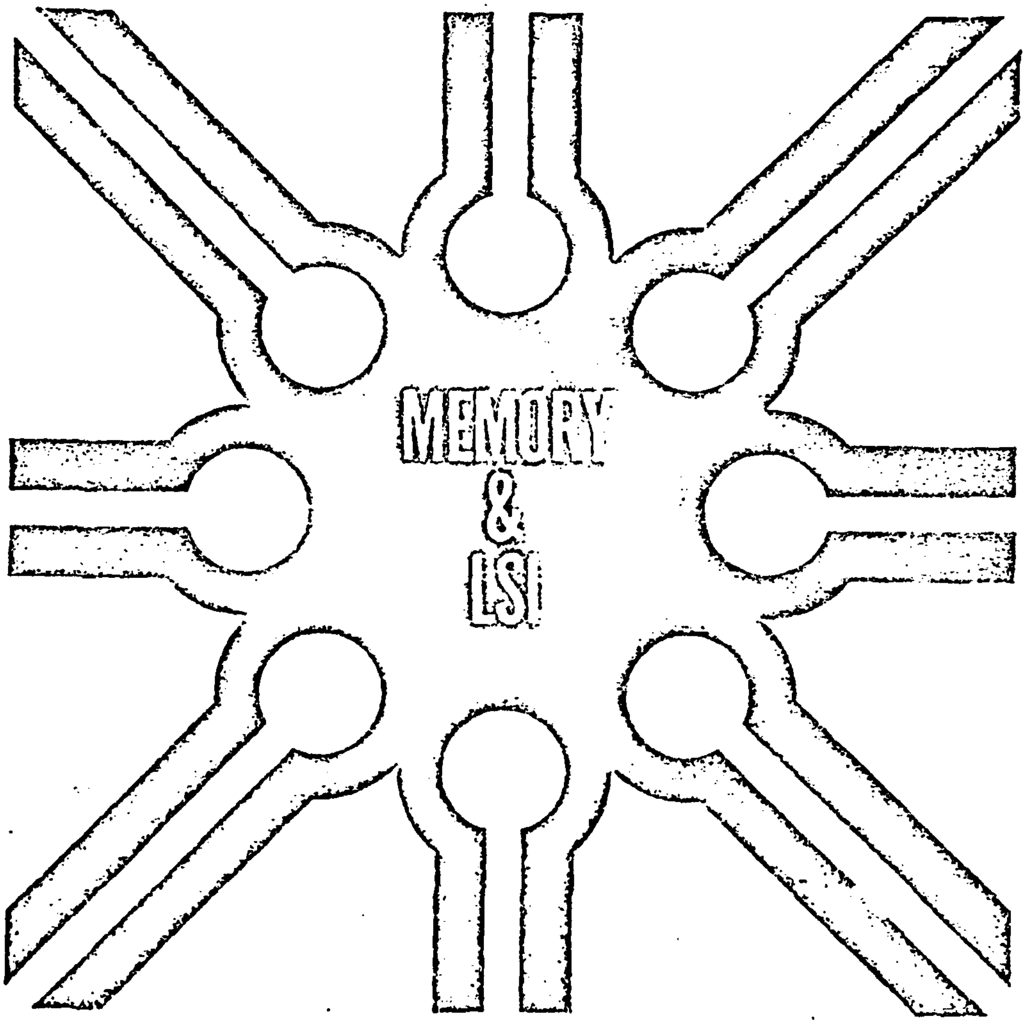
8. Joseph R. Herr, "Self-Checking Number Systems", Computer Design, June 1974, pp. 85-91.

9. Peterson and Weldon, Error-Correcting Codes, MIT-Press, 1972.

10. Shu Lin, An Introduction to Error-Correcting Codes, Prentice-Hall, 1970.

DIGEST OF PAPERS

# 1977 SEMICONDUCTOR TEST SYMPOSIUM



October 25–27, 1977
held at Cherry Hill, New Jersey

Sponsored by
IEEE Computer Society
and the
Philadelphia Section
of the IEEE

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

Additional copies of this digest available from:
IEEE Computer Society Publications Office
5855 Naples Plaza, Suite 301
Long Beach, California 90803

77CH 1261-7C

# IDENTIFICATION OF CAUSES OF PATTERN SENSITIVITY

D. D. Rinerson and A. Tuszynski

## Abstract

Pattern Sensitivity at the device level is attributable to deficiencies in circuit design and processing; at the system level, user misdeeds must be additionally taken into account. Meaningful testing for Pattern Sensitivity requires diagnostic expertise in all these areas. Though difficult to detect by hit and miss techniques, Pattern Sensitivity is deterministic in nature, can be tracked down by sufficiently sophisticated testing, and will be eventually eliminated through improvements in circuit design and process control.

## Introduction

Though the concept of Pattern Sensitivity (PAS) is somewhat elusive, Schmoo plots tell the story eloquently. Figures 1 and 2 show the results of tests performed on a suspect 4k-RAM. The Schmoo plot for "Column Disturb" differs significantly from that for "Galpat". Though each of the two programs checks the operation of every individual storage cell, the unit under test passes the Galpat test but fails the Column Disturb. The difference in the Schmoos is attributable solely to the difference in the data patterns generated by the respective programs. There is nothing accidental in these results. Once identified, Pattern Sensitivity can be reproduced on command, as demonstrated by the rerun tests shown in figures 3 and 4. Other tests show that the conslusions regarding the relative efficiency of particular programs can be extended to all devices of the same type and the same origin, but not necessarily to second-source parts.

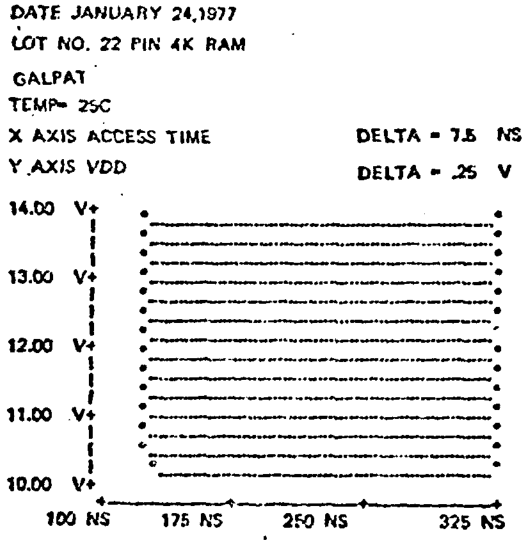


FIGURE 1 : GALPAT TAKEN 1/24/77 at 6:00 PM

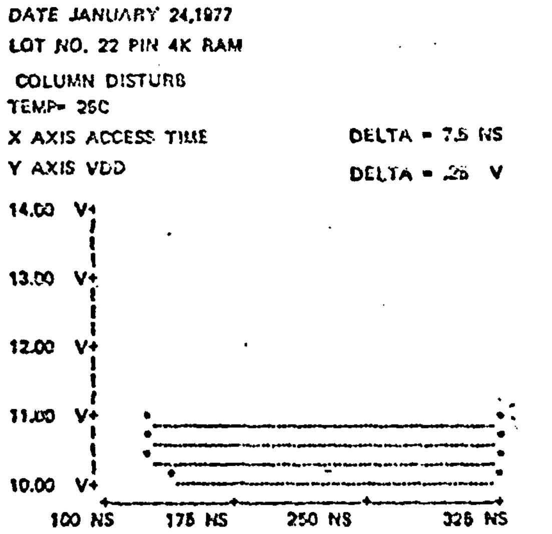


FIGURE 2 : COLUMN DISTURB TAKEN 1/24/77 at 7:30 PM

In spite of the complexities of LSI, there is actually no mystery in PAS. Evidently enough, the Galpat program does not generate the worst case pattern of the given type of device. The Column Disturb test is more severe and, incidentally, more efficient than Galpat - it takes 20 minutes to run a full TACC/$V_{DD}$ Column Disturb Schmoo but a full hour to do the same for Galput - yet, for all we know, it may not be severe enough. Only those programs which actually generate worst

Darrell Rinerson is a project engineer at the Sperry-Univac Corporation in Roseville, MN and a graduate student at the University of Minnesota in Minneapolis.

Al Tuszynski is an associate professor of electrical engineering at the University of Minnesota.

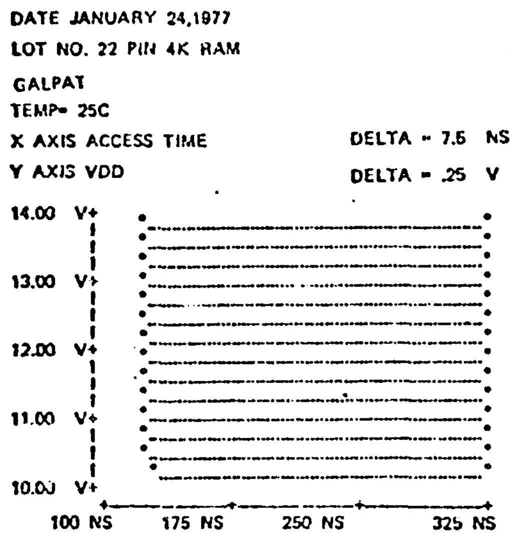


FIGURE 3 : GALPAT TAKEN 1/24/77 at 8:00 PM

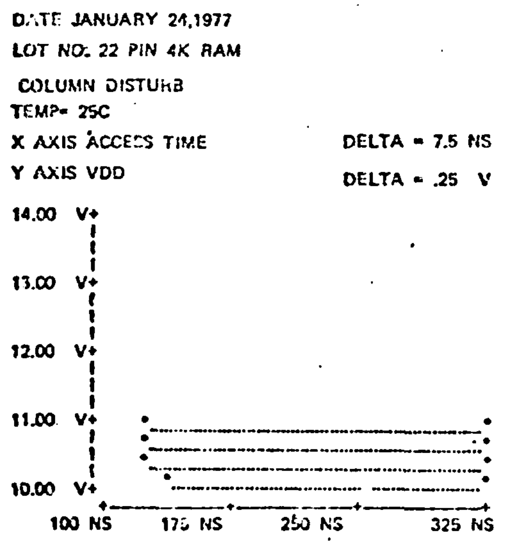


FIGURE 4 : COLUMN DISTURB TAKEN 1/24/77 at 9:30 PM

case data patterns are sufficiently severe. Good programs generate worst case patterns efficiently. To develop such programs, one must begin with an adverse analysis of the circuitry and then follow through with process diagnostics and exploratory testing.

Theoretical Speculations

Any deterministic system can be completely modeled by matrices A and B to be used as operators in equations which relate the state variables x to the input vector u as in

$$\dot{x} = A\,x(t) + B\,u(t) \qquad (1)$$

Unfortunately, in real-life LSI problems, the dimensions of matrices A and B are prohibitive while the elements $a_{ij}$, $b_{lk}$ are non-linear and only vaguely predictable. Consequently, one seldom employs the full state-space presentation in actual analysis of LSI circuits. Nonetheless, a few useful conclusions can be drawn from a free-wheeling interpretation of these equations.

Microprocessors, memories and most other LSI devices fall into the category of "continuous-time systems with discrete-time inputs". The transition equations of such systems[5] relate the state variable x to the input u through the matrix operators A and B as in

$$x(kT+\tau)=[e^{A\tau}]x(kT)+\int_{kT}^{kT+\tau}[e^{A(kT+\tau-v)}]Bu(v)dv \qquad (2)$$

where: T = clock period
and (kT + τ = t)

With this in mind, let us feed new data into the system at t = kT and let us observe the state of the system at t = kT + aT, where a is less than one, but greater than zero. Beginning at t = o with an initial state x(o), we get

$$x(aT)=[e^{aAT}]x(o)+\int_{o}^{aT}[e^{A(\tau-v)}]Bu(v)dv \qquad (3)$$

at the time of the first observation, and

$$x(\tau+aT)=[e^{aAT}]x(T)+\int_{T}^{T+aT}[e^{A(T+aT-v)}]Bu(v)dv \qquad (4)$$

at the time of the second.
Evidently then, the state of the system at t=aT depends on the initial state x(o) as well as the input u(o), while the state at t=T+aT depends on u(1), u(o) and t(o).

In other words, the output at t = T+aT depends not only on the data pattern fed into the system at t+T but also on the state of the system at t=T, and that depends in turn on the data input at t=o as well as the initial state x(o). If we now read the last output as either true or false only, then we may conceivably get "correct" or "incorrect" outputs in response to a fixed u(1) but varied u(o) or x(o). This constitutes either pattern or sequence sensitivity or both, depending on the interpretation of pertinent events.

The extent of the above dependence is directly related to the duration of the clock period T. The effect of history decays with time and consequently, as the period is gradually extended, x(T+aT) becomes virtually independent of u(o) and even more independent of x(o). Furthermore, its value approaches the extrapolated steady state level

$$\underset{T\to\infty}{\text{limit}}\ x(T+aT) = \underset{t\to\infty}{\text{limit}} \int_T^t \{e^{\Lambda(t-v)}\}Bu(v)dv$$

It follows then that one should never encounter dynamic Pattern and Address Sequence Sensitivity (PAASS) at relatively low speeds although one will invariably run into complications at excessive speeds. To avoid dynamic PAASS one must simply determine the worst case dynamic response in terms of data patterns, cell locations, clock skews or what have you, and then spec out the device accordingly. What to the eyes of the accidental observer appears to be random PAASS, is actually a deterministic demonstration of excessive speed. You can't run an 8MHz unit at 10MHz and blame your problems on PAASS.

The transmission matrices (A and B) of LSI devices are less sparse than those of equivalent discrete circuits. The difference lies of course in skew elements that model crosstalk. Naturally enough, if a device works at all, one expects that significant parasitic coupling is limited to neighboring devices. Thus, in an assembly of storage cells, one expects more crosstalk between, say,

$a_{ii}$ and $a_{i-1,\ i+1}$

than between

$a_{ii}$ and $a_{i-2,\ i+2}$

Particularly important are of course cumulative effects. Consider for example the bit-line-sensing arrangement introduced by TI a few years ago and adopted since then by many other manufacturers of dynamic memories (figure 5). The voltage difference $\Delta V$, read by the sense-amplifier, depends on 1) the charge in the addressed cell, 2) the charge in the reference cell and 3) the nodal voltages of the stand-by cells. The last of these contributions is of course a parasitic force which arises from conductive and non-linear reactive leakage. If the cells on the left hold many more zeros than those on the right (or vice versa), then the parasitics may become significant. Particularly objectionable are data patterns which have a single zero on one side but a full field of zeros on the other, or conversely, a single high on one side but a full field of highs on the other.

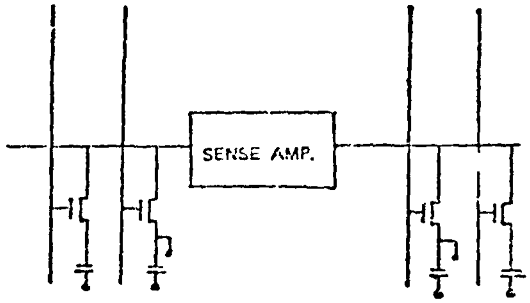


FIGURE 5: SIMPLIFIED SENSE AMP.

Finally, one must realize that most performance deficiencies arise from oversight on the part of the designer.

The first commercial static memories suffered from inadequate address separation, early "one transistor" RAMs had insufficient bit-line swing and a RAM of later vintage was fouled up by an illegal combination of transmission gates[6]. Interestingly enough, many such problems were detected by competitors a long time before they were corrected by the original vendors. Design flaws can be detected by bystanders since eyeball analysis is frequently more effective than CAD. One should bear that in mind when testing LSI devices.

Exploratory Testing

Although theoretical speculations yield clues rather than answers, clues can be developed into answers by exploratory testing. In the case of the 16 pin 4k-RAM, one suspects, for example, susceptibility to $V_{DD}$ noise and RAS/CAS delay ($t_{RLC}$) jitter. Either of these could lead to marginal performance and, therefore, pattern sensitivity. It is evidently advisable to perform a few experiments.

Susceptibility to $V_{DD}$ noise was investigated by means of the equipment shown in figure 6. Various AC signals were fed into the sense port of the regulator while "Column Disturb" Schmoo runs were being executed. Continuous and pulsed sine wave excitation was used to simulate high frequency perturbation of $V_{DD}$. Somewhat surprisingly, the 4k-RAMs withstood this challenge very well. Specifically, all devices subjected to these tests turned out to be immune to 50MHz signals of 2.4 volts peak-to-peak amplitude. Nonetheless, an entirely different story emerged from low frequency tests, conducted with triangular and trapezoidal waves of various slew rates. Dependence on synchronization of the "noise" to the system clock was expected and indeed confirmed. Various weaknesses were observed on devices procured from four different sources. Most interesting of these was sensitivity to write-read supply-line differentials. A test with 10.8 volts for write but 13.2 volts for read rejected all devices of one manufacturer but passed every one of the devices supplied by the other vendors. The units which failed the test worked quite well on smooth margins of $V_{DD}$ and equally well when written at 13.2 but read at 10.8 volts.

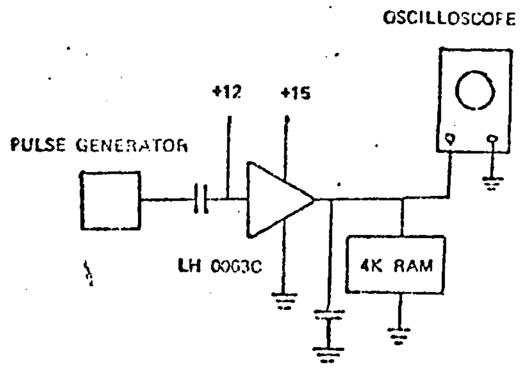


FIGURE 6 : EQUIPMENT SETUP

The above mentioned test - sometimes referred to as the "Voltage Bump" test - does detect supply line sensitivity very effectively. It also facilitates identification of weak circuit elements. While it is almost customary to point to the sense amplifier as the principal victim of $V_{DD}$ perturbations, we have found, for example, that the internal timing chain is equally weak. Many devices which passed the Voltage Bump test at nominal $t_{RLC}$ failed on the margins of that parameter; they worked well with smooth supply lines of 10.8 and 13.2 volts respectively, but not with the equivalent Voltage Bump. It is evident, therefore, that the timing and noise specifications of memories require far more attention than they have received in the past.

Follow-up

With the results of extensive theoretical and experimental studies in hand, it is relatively easy to write programs which surpass in effici. ıcy any assortment of Marches, Galpats and Disturbs. One is convinced that PAS suspects can be and will be caught practically without fail. Nonetheless, one knows that field failures occur far too often. Worse still, the "rejects" frequently check out as good devices when retested in the laboratory. Evidently, there must be a lost link somewhere.

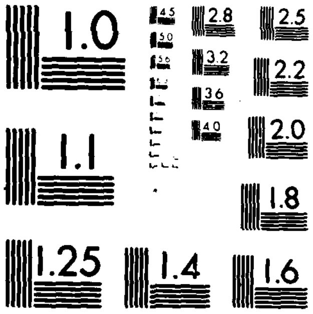
Random PAS is certainly a possibility. Accidental coupling between physically distant and electrically unrelated storage cells cannot be tracked down by deterministic means. The detection statistics of such faults are absolutely incredible. Fortunately, the likelihood of such faults is very small in the first place, certainly much smaller than the rate of field failures. We must look somewhere else for an explanation. There may exist some overall system problems. Worst case combinations of noise, signal delays and supply perturbations may just suffice to upset a few marginal devices. Poor understanding of latent characteristics of LSI devices is the main cause of such problems. Current spikes during "refresh" exceed the average consumption by an order of magnitude and may induce significant noise on supply lines, in ground wires, and at the signal ports. The refresh mechanism of memories may in itself be susceptible to various kinds of noise. In the presence of noise, some memories work in write-delay-read cycles, but fail in write-refresh-read operations. Especially treacherous, though, is the latent sensitivity of the CAS/RAS delay. In some 16k-RAMs, the range of $t_{RLC}$ is reduced from 60 to 30 [illegible] Voltage Bump transients. All in all, [illegible]

AD-A110 263    MINNESOTA UNIV  MINNEAPOLIS  DEPT OF ELECTRICAL ENGIN--ETC  F/6 9/5
ESSENTIAL PATTERN AND SEQUENCE SENSITIVITY IN SEMICONDUCTOR MEM--ETC(U)
JUL 80   A TUSZYNSKI                                    N00014-78-C-0741

UNCLASSIFIED                                                                    NL

2 of 2
AI
A110263

END
DATE
FILMED
3-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

not surprising that field failures do occur. Some of these may be attributable to unidentified PAS, but many more arise from incomplete information on the mundane characteristics of semiconductor memories.

## Conclusion

Flaws in circuit design, chip layout, and processing cause PAS as well as all other faults. Circuit design problems lead the pack, but they can be identified by theoretical and experimental analysis. Extensive exploratory work is essential, but experimental results must be treated with caution. They portray complex phenomena and frequently apply only to devices of common origin. For example, the Voltage Bump test observed to fail a group of devices with 10.8 to 13.2 volt write to read ramps works in reverse on memories from other families.

Data sheets occasionally conceal more than they reveal. Specifications are meaningless in the absence of noise limits. Timing tolerances depend on supply line noise. The refresh mechanism of some devices suffers from defects other than plain leakage. All such problems lead to pattern sensitivity, but there is nothing mysterious about them. Pattern Sensitivity is essentially deterministic in nature; it can be identified and will be eventually eliminated at the design level by better circuit and process models.

## Acknowledgements

Thanks are due for help from Mr. O. H. Lindberg, Manager, Microelectronics Division, Naval Ocean Systems Center, San Diego, California.

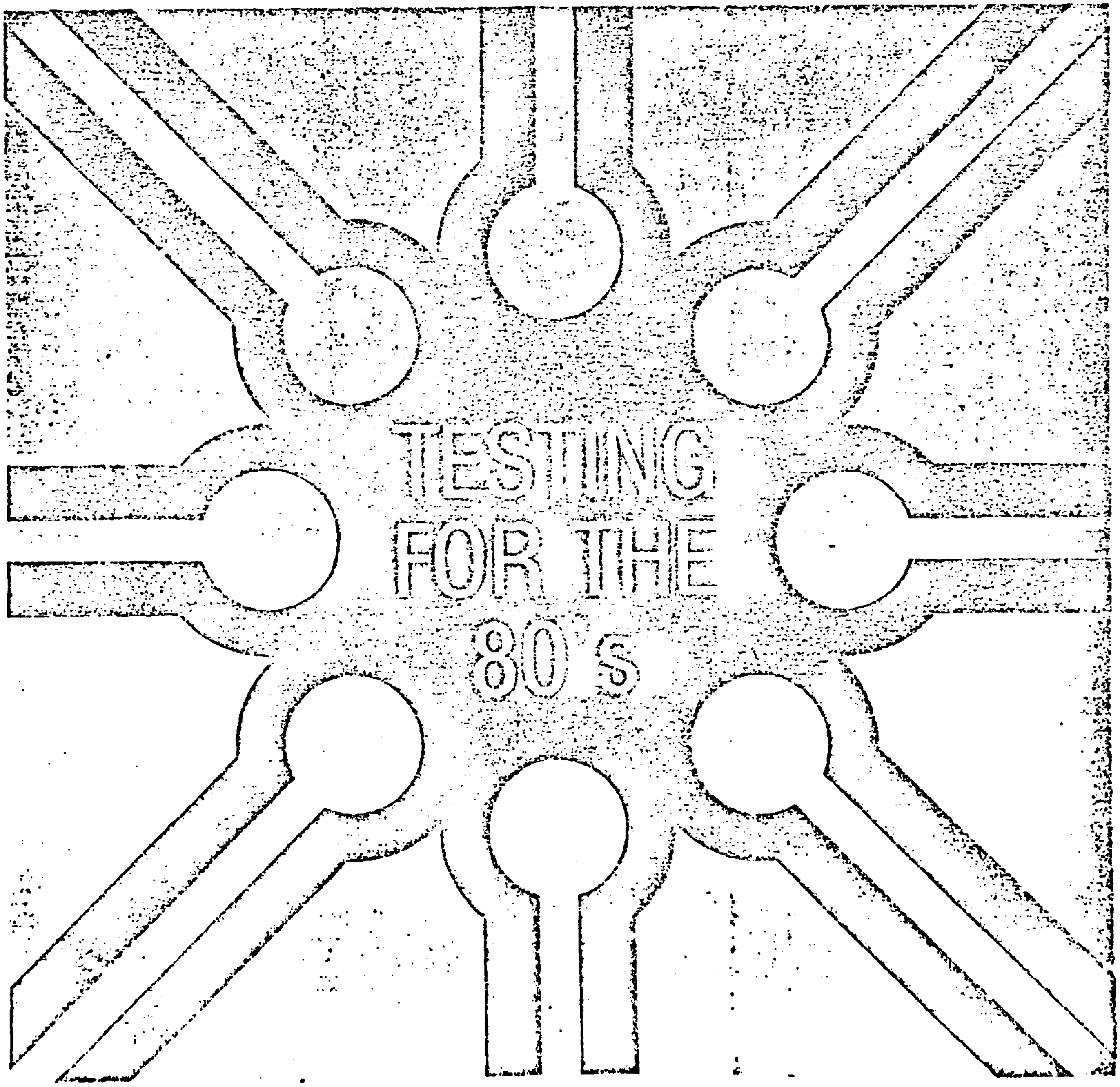
## References:

1. George P. Nelson, "Microprocessor Testing and Performance Qualifications:, International Symposium on Military & Industrial Microprocessor Systems; San Diego, Co., June, 1975.
2. A. C. L. Chiang and R. Standridge, "Pattern Sensitivity on 4k RAM Devices", Computer Design, Feb. 75, pp. 88-90.
3. R. Hickmeister and A. C. L. Chiang, "Microprocessor Test Technique Reveals Instruction Pattern Sensitivity", Computer Design, Dec. 75, pp. 81-88.
4. J. Henk de Jonge and A. J. Smulders, "Moving Inversion Test Pattern Is Thorough, Yet Speedy", Computer Design, May 76, pp. 167-173.
5. Hebert Freeman, Discrete-Time Systems, John Wiley, 1965.
6. W. S. Richardson, "Diagnostic Testing of MOS Random Access Memories", Solid State Technology, March 1979, pp. 31-34.

DIGEST OF PAPERS

# 1980 TEST CONFERENCE

November 11, 12 & 13, 1980
sponsored by the IEEE Computer Society,
Test Technology Committee
and the Philadelphia Section of the IEEE



IEEE COMPUTER SOCIETY
INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

# PSEUDO ANALOG TESTING

Daniel P. Burbank and A. Tuszynski

University of Minnesota

## Abstract

Pseudo Analog Testing circumvents difficulties encountered when attempting to diagnose large monolithic circuits. Rather than swamp a high-impedance network with the input capacitance of a test probe, one relies on indirect but credible results. To estimate the behavior of a critical node B, one observes the stream of data which flows from the output port, while exploratory analog perturbations are applied to a robust "upwind" node A. Performance margins and other bits of information are deduced from the magnitudes of those perturbations which convert true ONES into false ZEROS and true ZEROS into false ONES.

## Background

Examining the behavior of new circuits, one does not stop at the I/O pads. Internal nodes are examined in pursuit of analog information that either validates expectations or divulges the causes of malfunctions. Unfortunately though, this is easier said than done. Power economy being a fundamental prerequisite of Large Scale Integration (LSI), one is invariably faced with high-impedance regimes, especially in MOS-LSI. There might be some stand-by current in the output ports, but no such waste will be found in the internal branches of well-designed chips. Pre-charge or CMOS techniques are the rule of the day. A typical subset of components often resembles a stand-alone capacitor, over a substantial part of the operating cycle. The magnitude of the equivalent capacitor may be quite small at that, much less than a picofarad, more often than not. The monitoring problem is, therefore, severe. Conventional oscilloscope probes are entirely inadequate; they will literally swamp the device under test, altering the local voltage/current relationship or, worse still, upsetting the normal mode of operation of the entire circuit. Relief can be gotten -at the cost of compromises- in some but not all instances. More exotic measures must be brought into play to deal with the general case.

At least three distinct techniques can be mustered when the electrical characteristics of high-impedance networks are to be measured at regular clock rates. First-off, there is the well known Low-Input-Capacitance Voltage-Follower [1-3] of Figure 1. Positive feedback is employed in this instrument to offset the real and tangible capacitances of the probe and fixture with an artificial negative capacitance

$$C_n = -C_{fd}(K1*K2 - 1) \tag{1}$$

There is a flaw in this scheme. Being feedback dependent, $C_n$ varies with frequency. If f1 and f2 be the cut-off frequencies of the Buffer and Amplifier respectively, then

$$C_n = - C_{fd}\left[\frac{K1*K2}{(1 + jf/f1)(1 + jf/f2)} \quad 1\right] \tag{2}$$

and therefrom comes a sobering conclusion: the probe looses its effectivenese at f1/10 or f2/10, whichever is smaller.

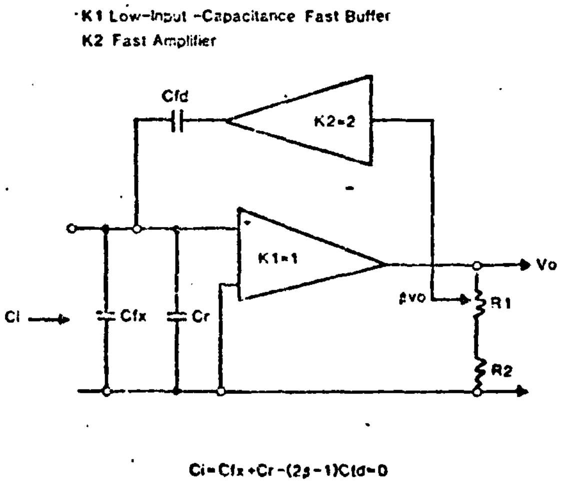


Fig. 1 A Low Input-Capacitance Voltage-Follcver

A second, entirely different, technique relies on bombardment with a narrow beam of electrons to provide resolution in space, time and voltage of one micron, one nanosecond and 20 millivolts. Unfortunately though, the impressive frequency

*Work leading to this article was funded by the Office of Naval Research under contract No. 00014-78-C-0741 monitored by Dr. Joel M. Morris.

**Daniel Burbank is now with the Honeywell Solid State Circuits Center in Minneapolis, MN.

CH1608-9/80/0000-0126$00.75 © 1980 IEEE

response of the so called Stroboscopic Scanning Electron Microscope [4-6] is attributable entirely to sampling. This means that only periodic waveforms are reproduced, one-shot phenomena and noise are suppressed. Furthermore, one has to put-up with the many inconveniences of vacuum equipment and, worse still, the required equipment is not available commercially, just yet [7,8]. That is why yet another technique -Pseudo Analog Testing- does acquire the status of a viable option, in spite of its own peculiar deficiencies.

## The Pseudo Analog Technique

Pseudo Analog Testing [9] circumvents the need to probe sensitive nodes. The regular flow of data is obstructed by quasi-static analog signals, injected into a relatively robust part of the circuit, by means of a laser, a current source or a voltage generator [10]. The characteristics of a sensitive or inaccessible element are deduced from the pattern of errors brought on by the analog perturbations. In Figure 2, block C stands for the sensitive or inaccessible node, the characteristics of which are to be determined; block B represents a "robust" neighbor. Data flows from A, through B and C, to D. The analog perturbations injected at B affect C and C only. The performance margins of C are measured in terms of magnitudes and polarities of perturbations which cause errors.

This sounds far fetched and it is, often enough. One can name, however, applications in which Pseudo Analog Testing does make sense. A particularly convincing example is furnished by the bit line diagnosis of dynamic RAMS.

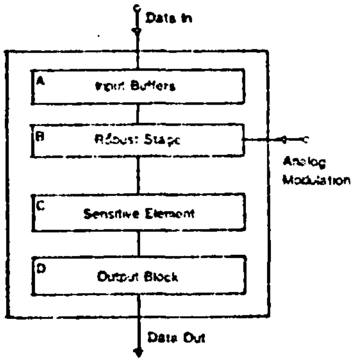


Fig. 2 Analog Perturbation of Digital Signals

## Diagnostics of Dynamic RAMS

Verifying the quality of large chips, one takes a two-pronged approach: 100% testing for hard failures on one hand, but sampling and selective pursuit of soft errors on the other. Soft errors come in many varieties. Though malfunctions induced by alpha particles [11-13] have held the limelight in the seventies, other phenomena -processing flaws and design mistakes among them- can be more disruptive [14-16]. All kinds of obscure secondary effects frustrate the designer of dynamic RAMS [17-19]. Fortunately though, many a perplexing problem can be unravelled at the bit-line level (Figures 3 and 4).

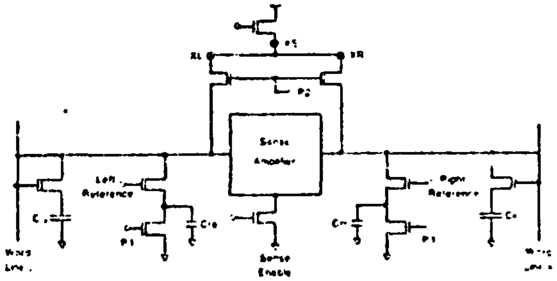


Fig. 3 Essential Elements of the Bit Line

Pertinent to our discussion of bit-line circuitry are six sets of components:
1. Storage capacitor $C_j$ on the left and reference capacitor $C_{rr}$ on the right of the Sense Amplifier
2. Gating transistors $Q_j$ and $Q_{rr}$
3. Reset transistor $Q_{rr}$
4. Sense Amplifier [20] transistors $Q_{sl}$ and $Q_{sr}$
5. Sense-Enable transistors $Q_{el}$ and $Q_{er}$
6. Auxiliary transistors $Q_{al}$ and $Q_{ar}$
Not to be overlooked are the parasitic capacitances of the bit line, simulated by lumped capacitors $C_{pl}$ and $C_{pr}$ in the block diagram of Figure 3. These distributed capacitances have a crucial bearing on the performance margin of the RAM. They should be reduced to zero, but practical limits are set by the state of the art in processing technology.

A regular READ cycle includes the following events:
1. The top plate of the reference capacitor is grounded and the Sense-Enable transistors are cut-off, while the auxiliary transistors are driven into a high-impedance state.
2. Both sections of the bit line are precharged to almost $V_{DD}$
3. The gating transistors are enabled and the voltage on the right hand side of the Sense Amplifier is reduced to

$$V_r = \frac{C_{pr}}{C_{pr} + C_{rr}}(V_{DD} - \Delta V_{DDr}) \qquad (1)$$

while that on the left either remains relatively constant at

$$V_l(H) = V_{DD} - \Delta V_{DDl} \qquad (2)$$

or else degrades to

$$V_l(L) = \frac{C_{pl}}{C_{pl} + C_j}(V_{DD} - \Delta V_{DDl}) \qquad (3)$$

depending on the data stored in $C_j$.

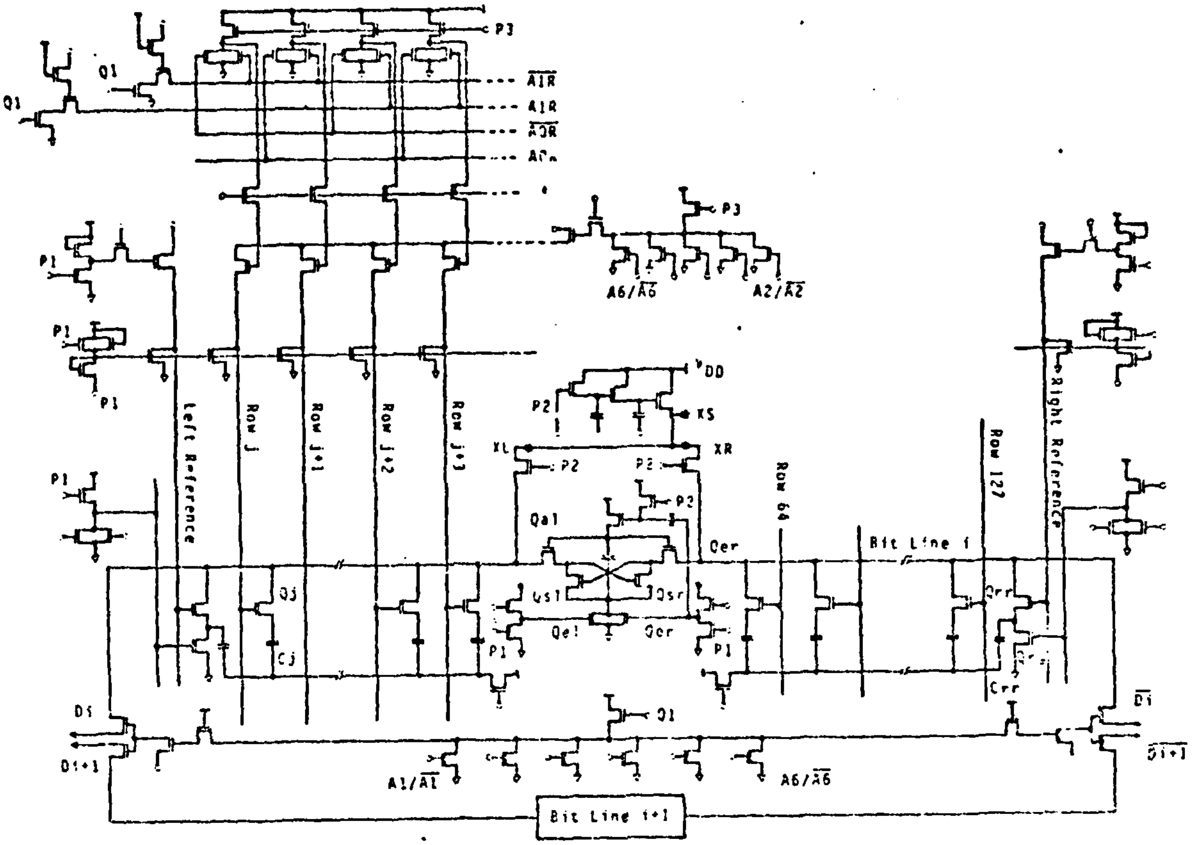


Fig. 4 Bit-Line Circuitry a la MK-4116

The $\Delta V_{DD}$ terms in the above equations account for the degradation of the bit line voltage by precharge imperfections. What really matters, however, is the difference between the left and the right rather than the absolute value of the bit-line voltage. We write, accordingly,

$$\Delta V = \Delta V_{DDl} - \Delta V_{DDr} \tag{4}$$

to get a simple symbol for the precharge error. It would appear that at least two other complications must yet be acknowledged: The offset voltage of the amplifier and the stagger of the bit-line capacitances. The former is denoted $\varepsilon$ while the latter is written as

$$\Delta C_p = C_{pl} - C_{pr} \tag{5}$$

One might add that, contrery to the design intent, the parasitic bit-line capacitances $C_{pl}$ and $C_{pr}$ are much larger than the storage cell capacitors. In spite of that, there is surprisingly little merit to equation 5; its practical consequences are negligible. It is therefore permissible to write:

$$C_{pl} = C_{pr} = C_p \tag{6}$$

$$\frac{C_j}{C_{pl} + C_j} = \frac{C_j}{C_p} \tag{7}$$

$$\frac{C_{rr}}{C_{pr} + C_{rr}} = \frac{C_{rr}}{C_p} \tag{8}$$

The effective differential voltage ($\delta V$) that appears at the input terminals of the Sense Amplifier can now be written as:

$$\delta V(H) = V_l(H) - V_r + \varepsilon \tag{9a}$$

$$= (C_{rr}/C_p)V_{DD} + \varepsilon - \Delta V \tag{9b}$$

$$\text{or} \quad \delta V(L) = V_l(L) - V_r \tag{10a}$$

$$= -[(C_j - C_{rr})/C_p]V_{DD} + \varepsilon - \Delta V \tag{10b}$$

Comparison of equation 9 with 10 shows that best results will be obtained with

$$C_{rr} = C_j - C_{rr} \tag{11a}$$

i.e.

$$C_{rr}/C_j = 1/2 \tag{11b}$$

Verification of this ratio is, accordingly, the first objective of bit-line diagnosis. Naturally enough, one also wants to determine ε and ΔV.

To implement the pseudo-analog technique, one attaches the differential bit-line supply-network (Figure 5) to nodes $X_s$, $X_l$ and $X_r$ of the RAM'under test (Figure 4). The old metal filaments are destroyed by a current surge from a 50 microfarad capacitor. A new degree of freedom is thus secured. The right section of the bit line can be offset relative to its left-hand counterpart by an arbitrary voltage.

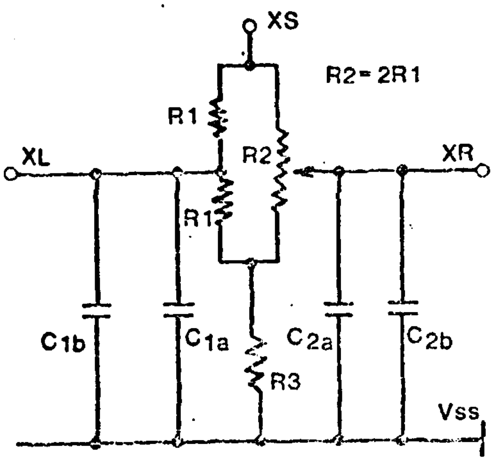


Fig. 5 Differential Bit-Line Supply

The RAM under test is operated at normal speed, during a typical PAT exercise. Data is written, refreshed and read through the regular I/O and control pads of the chip; nothing is allowed to obstruct the flow of digital data. Analog modulation, on the other hand, is injected through three microprobes; the amplitude and polarity of this perturbation is adjusted -in pseudo-static style- till true ONES are converted in false ZEROS or true ZEROS are transformed into false ONES. The overall performance margins become immediately apparent; resolution into individual error-components is accomplished through judicious variation of addresses and data patterns, followed by selective blockage of gating transistors. Sequential bit-failure-mapping, a la Figure 6, yields both the absolute value and the spread of the $C_J/C_{rr}$ ratio.

Some pseudo-analog tests can be performed without recourse to drastic measures. The leakage characteristics of storage cells -to name one prominent parameter- can be studied through the mechanism of the extra-long refresh cycle. One simply finds out how long it actually takes -at any particular temperature- to discharge a storage capacitor. Again, one gets surprisingly interesting absolute and relative values.

The significance of relative values is not to be overlooked. When it comes to soft errors, relative values are more important than their absolute counterparts. Mild deviations from the norm are particularly ominous. They should be examined in detail. Physical locations and electrical links must be noted. A top logical or functional pattern may emerge and thus reveal a source of soft errors.

Conclusion

Analog instrumentation for VLSI diagnostics is still in its infancy. As yet, there are no universally satisfactory techniques; one fits the method to the job, settling for all kinds of limitations and inaccuracies. Pseudo Analog Testing is slow and destructive, but it yields results where everything else fails.

References

1. Hugo J. DeMan, Rene A. Vanparys & Roger Cuppens, "A Low Input Capacitance Voltage Follower in a Compatible Silicon-Gate MOS Bipolar Technology", IEEE J.S.S.C. Vol. SC-12, pp. 217-225, June 1977.
2. George Erdi, "A 300 V/μs Monolithic Voltage Follower", IEEE J.S.S.C. Vol. SC-14, pp. 1059-1065, December 1979.
3. Kam H. Chen and Robert G. Meyer, "A Low-Distortion Monolithic Wide-Band Amplifier", IEEE J.S.S.C. Vol. SC-12, pp. 685-690, December 1977.
4. G. Y. Robinson, "Stroboscopic Scanning Electron Microscopy at Gigahertz Frequencies", The Peview of Scientific Instruments, Vol. 42, pp. 251-255, February 1971.
5. Eckhard Wolfgang, Rudolf Lindner, Peter Fazekas and Hans-Peter Feuerbaum, "Electron Beam Testing of VLSI Circuits", IEEE J.S.S.C. Vol. SC-14, pp. 471-481, April 1979.
6. Hiromu Fujioka, Koji Nakamae and Katsumi Vra, "Function Testing of Bipolar IC's and LSI's with the Stroboscopic Scanning Electron Microscope", IEEE J.S.S.C. Vol. SC-15, pp. 177-183, April 1980.
7. Editorial, "E-Beam Measures Chip Voltages, Cuts Probe Capacitance", Electronics, p. 162, July 17, 1980.
8. John Gosh, "Electron Beam Harmlessly Probes High-Density Chips", Electronics, pp. 65-66, July 31, 1980.
9. Daniel P. Burbank, "Pseudo Analog Testing of Dynamic Memories", M.S. Thesis, University of Minnesota, Minneapolis 1979.
10. J. M. Gaworecki, N. G. Thoma and A. J. Wager, "Sense Signal Characterization and Test Aid for One-Device Dynamic Memory Arrays", IBM Technical Disclosure Bulletin, Jan. 8, 1979.
11. T. C. May and M. H. Woods, "A New Mechanism for Soft Errors in Dynamic Memories", Reliability Physics 1978, IEEE Contract No. 79CH1294-5PHY, 6 pages.
12. J. A. Woolley, L. E. Lamer, N. H. Stradley and D. H. Harshbarger, "Low Alpha Particle Emitting Ceramics", IEEE Trans. Components, Hybrids & Manufacturing Technology, Vol. CHMT, pp. 388-390, December 1979.
13. Alan James Levandowski, "Effects of Alpha Particles on Dense Storage Devices", M.S. Thesis, University of Minnesota, July 1979.

14. J. Reese Brown Jr., "Timing Peculiarities of Multiplexed RAMS", Computer Design, pp. 85-92, July 1977.

15. T. C. Lo and Mark R. Guidry, "An Integrated Test Concept for Switched-Capacitor Dynamic MOS RAMS", IEEE J.S.S.C. Vol. SC-12, pp. 693-703, December 1977.

16. John P. Hayes, "Detection of Pattern-Sensitive Faults in Random Access Memories", IEEE Trans. on Computers, Vol. C-24, pp. 150-157, February 1975.

17. Osaku Kudoh, Makoto Tsurumi, Hiroshi Yamanaka and Toshio Wada, "Influence of Substrate Current on Hold-Time Characteristics of Dynamic MOS IC's", IEEE J.S.S.C., Vol. SC-13, pp. 235-238, April 1978.

18. D. D. Rinerson and A. Tuszynski, "Identification of Causes of Pattern Sensitivity", IEEE 1977 Semiconductor Test Symposium, pp. 166-170.

19. A. Tuszynski, "Self-Learning Machines Applied to the Testing of Semiconductor Memories", IEEE Autotescon 78, pp. 248-256.

20. Klinton Kuo, Nori Kitagawa and Phil Drayer, "Sense Amplifier Design is Key to 1-Transistor Cell in 4,096-bit RAM", Electronics, pp. 116-121, September 13, 1973.

21. James E. Coe and William G. Oldham, "Enter, the 16,384-bit RAM", Electronics, pp. 116-121, February 19, 1979.

22. C. N. Ahlquist, J. R. Breivogel, J. T. Koo, J. L. McCollum, W. L. Oldham and A. L. Renninger, "A 16K Dynamic RAM", IEEE 1976 ISSCC. pp. 128-129.

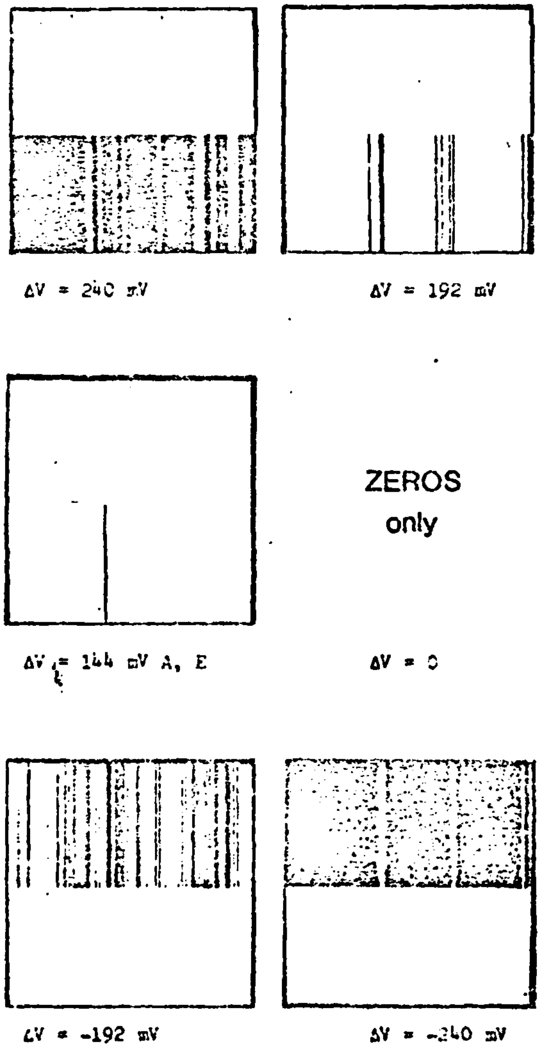


Fig. 6a: Full Field of ONES

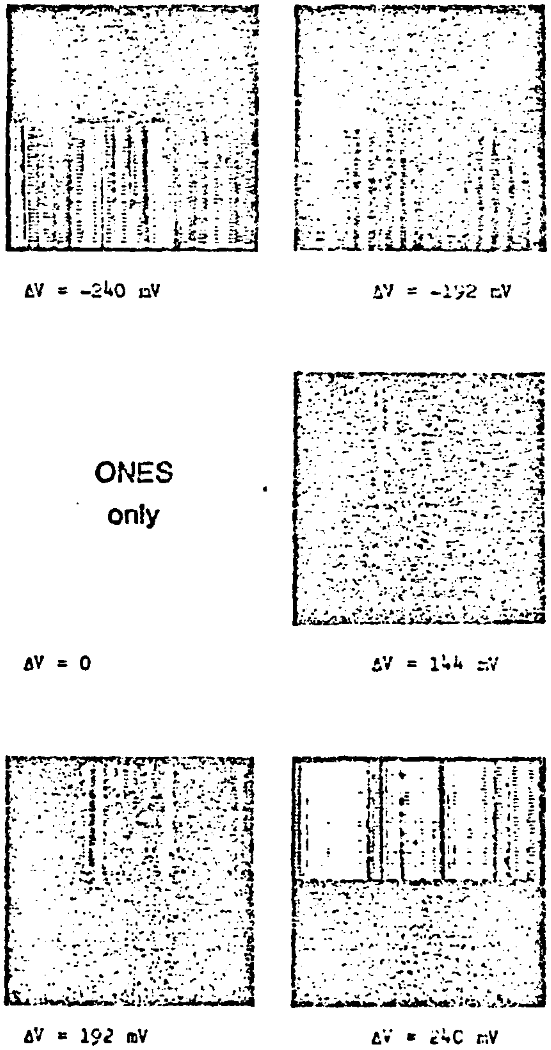


Fig. 6b: Full Field of ZEROS

DATE
FILMED

8